André Simão Pereira Catela

Master in Computer Science

Epik Platform - Design and Development of Interactive Puzzles as an Educational Activity

Dissertation submitted in partial fulfillment
of the requirements for the degree of

Master of Science in
Computer Science and Informatics Engineering

Adviser: Carmen Morgado, Assistant Professor, NOVA University of Lisbon

Co-adviser: Fernanda Barbosa, Assistant Professor, NOVA University of Lisbon

Examination Committee

Chairperson:
Raporteurs:
Members:

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

March, 2019

Epik Platform - Design and Development of Interactive Puzzles

This document was created using the (pdf)LaTeX processor, based in the "novathesis" template[1], developed at the Dep. Informática of FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Acknowledgements

Em primeiro lugar gostaria de agradecer às minhas orientadoras durante a realização da dissertação, Professora Carmen Morgado e Professora Fernanda Barbosa, por todo o apoio, conselhos, sugestões e dedicação durante todo o processo.

Aos meus pais, gostaria de agradecer pelo enorme apoio e por todos os esforços que fizeram para me poder oferecer todas as condições que me permitiram frequentar e terminar o curso. À restante família, gostaria também de agradecer por todo o apoio, motivação e preocupação durante a realização do curso.

Por fim, agradecer aos melhores amigos que a faculdade me podia ter oferecido por todo o apoio nos bons e maus momentos deste percurso, quer a nível académico como pessoal, e sem os quais não teria sido possível terminar o curso: Duarte Cruz, Francisco Cardoso, João Santos e Ricardo Fernandes e Tiago Castanho.

A todas as pessoas aqui mencionadas que influenciaram o meu percurso académico no Mestrado Integrado em Engenharia Informática, um muito obrigado.

Abstract

Nowadays, many types of games are used as an educational tool, capturing more easily students attention by keeping them interested in the lectured topics, which accelerates the learning process and provides collaborative learning in an entertaining way. Puzzle games in specific are one of the most used games for educational purposes due to its variety of formats, themes, and logics to solve them, having an adjustable difficulty according to students capacity of problem solving, with the view of developing these capacities.

Despite this gain of popularity, there are not a lot of options regarding computacional platforms to develop educational games. The Epik Platform consists in a web-based framework dedicated to the management of didactic contents and development of educational games. With this platform, the creation and edition of games turned to the educational environment is quite simplified, which makes it a good option for this purpose [1]. However, this platform's games only include quiz type activities of multiple choices, true or false and matching, not offering much variety for the users to work with.

Having this in mind, the development of interactive puzzle activities of different types, which is the main objective of this thesis, will allow the extension of the range of activities available in the set of games at the Epik platform. To be able to develop the interactive puzzle activities, a study must be done so that the right framework is selected according to the needs and limitations of development. With the addition of this new type of activity, the users will be able to develop their own interactive puzzles according to the three different puzzle games included, each one having a specific development environment for it. Overall, the platform will now be able to provide games performing a group of activities of different kinds in the same game, making it more appealing for the users.

At a global level, this dissertation is inserted on the project 'Restructure, Flexibilize and Update the Epik Platform', which aims to reimplement, extend and restructure features of the Epik platform, whose development was made for a previous master thesis by a student (Bruno Sampaio) [2].

Keywords: Educational Games; Puzzles; Epik platform; Interactive activities.

Resumo

Hoje em dia, muitos tipos de jogos são usados como ferramentas educacionais, facilitando a captura de atenção dos alunos, mantendo-os interessados nos temas leccionados, o que permite acelerar o processo de aprendizagem e proporcionar aprendizagem colaborativa de uma maneira divertida. Os puzzles em específico são um dos jogos mais utilizados para fins educacionais devido à sua variedade de formatos, temas e lógicas diferentes para os resolver, tendo dificuldade ajustável de acordo com as capacidades de resolução de problemas dos estudantes, com vista ao desenvolvimento dessas mesmas capacidades.

Contudo, apesar deste acréscimo de popularidade, não existem muitas opções de plataformas para o desenvolvimento de jogos educacionais. A plataforma Epik consiste numa *web-based framework* dedicada à gestão de conteúdos didáticos e ao desenvolvimento de jogos educacionais. Nesta plataforma a criação e edição de jogos de vertente educacional são bastantes simplificadas, tornando-a uma boa opção para este propósito [1]. No entanto, o único tipo de jogos incluído na plataforma são atividades do tipo *quiz*, como escolha múltipla, verdadeiro ou falso e correspondências, oferecendo pouca variedade.

Tendo em vista este aspecto da plataforma, o desenvolvimento de atividades interativas do tipo puzzle irá permitir ampliar o número de atividades disponíveis no conjunto de jogos da plataforma Epik. Para possibilitar o seu desenvolvimento, deverá ser feito um estudo para que a *framework* selecionada esteja de acordo com as necessidades e limitações do desenvolvimento. Com a adição deste novo tipo de atividades, os utilizadores poderão desenvolver os seus próprios puzzles interativos, de acordo com os três tipos de puzzles incluídos, existindo um ambiente próprio para o desenvolvimento de cada um. No seu todo, a plataforma será capaz de providenciar jogos com grupos de atividades de diversos tipos num mesmo jogo, tornando-a assim mais apelativa para os utilizadores.

A nível global, esta dissertação encontra-se inserida no projecto 'Restructurar, Flexibilizar e Inovar a Plataforma Epik', que tem como objectivo reimplementar, expandir e restruturar as funcionalidades da plataforma Epik, desenvolvida numa anterior tese de mestrado (Bruno Sampaio) [2].

Palavras-chave: Jogos educativos; Puzzles; Plataforma Epik; Atividades interativas

Contents

List of Figures

List of Tables

Acronyms

AR	Augmented Reality.
CSS	Cascading Style Sheets.
Epik	Edutainment by Playing and Interacting with Knowledge.
HTML	HyperText Markup Language.
HTTP	Hypertext Transfer Protocol.
IDE	Interactive Development Environment.
JS	JavaScript.
JSON	JavaScript Object Notation.
MVC	Model-View-Controller.
PDF	Portable Document Format.
PHP	Hypertext Preprocessor.
UI	User Interface.
VB	Visual Basic.
XML	Extensible Markup Language.

Chapter 1

Introduction

The presented chapter includes sections to introduce the topic of this dissertation, motivating for the work to be developed and the reasons why it is going to be realized, explaining the context in which it is inserted in the Epik Platform and in the project 'Restructure, Flexibilize and Update the Epik Platform', listing the main objectives for this dissertation and summarizing the content of the document.

1.1 Motivation

Learning standards used in education might not always be the best way to achieve knowledge. Moreover, games with educational strand are starting to become more popular among schools and as learning tools in general, as many types of games may be used in an educational environment [3]. They have long been employed as a means of education, as for example noblemen of the Middle Ages learned strategies of war using the ancient game of chess [4]. By using games as an educational tool, it is easier to capture students' attention and keep them interested, accelerating the learning process and providing collaborative learning in an entertaining way [5]. Students learn while playing games, teaching them goals, rules, adaptation and problem solving [4], while developing critical thinking skills, creativity, teamwork and good sportsmanship, creating positive memories of learning [5]. Furthermore, collaboration and social interaction during the games makes them more interesting and challenging to play, while it helps developing group work abilities and interpersonal skills [4].

Digital game-based learning is a novel approach in the area of universities and lifelong learning, and the search for new positioning of the universities in the changing setting of education: gaming is becoming a new form of interactive content, worthy of exploration [6]. Since the appearance of computers and internet, countless studies were made about

its influence and use in education, once they can provide an extra source of learning and convert learning into a more collaborative, interactive and dynamic process. With the advances of technology, nowadays the students' generation is fascinated by the gaming world, mostly because of the competition and the teamwork involved [3]. Scientists suggest that learning comes as the result of the environment that foster learning rather than as a result of teaching. Today's students look upon technology as an integral part of their life as it has been part of their learning since early childhood, which creates the need to offer a variety of different knowledge presentations and create opportunities to apply this knowledge within a virtual world, thus supporting and facilitating the learning process [6].

Despite the potential benefits of educational games, the e-learning solutions available most of the times offer boring and poorly structured learning materials that focus more on the technology instead of instructional support [6]. The potential effectiveness of these solutions largely depends on the pedagogical design behind the game and to what extent this design is aligned with the requirements of specific educational situations [7]. Moreover, there are not a lot of platforms with the aim of developing interactive educational games, and the existing ones, even when made for teachers, are not easy to use if the user is not an experienced programmer and are hard to adapt to individual courses and require much time for development [7]. The means of communication and collaboration between students and teachers are the core of education, allowing the transmission of knowledge and importance of teamwork, sense of mutual help, among others. However, these means are still the biggest obstacle in terms of integration of interactive educative activities, since the solutions available provide communication which sometimes is limited, not allowing to communicate and collaborate in real time. As a result, competition and teamwork development, which normally contribute to an increase in motivation for students, are discarded [8]. Furthermore, these games most of the times don't allow to store the information about the players performance, which is crucial for a teacher to evaluate or help the students according to its performance [9].

Having this in mind, the Epik Platform was developed as an attempt to solve these problems. It consists of a graphic tool with the aim of developing and managing educational games as a set of interactive activities which allow cooperative learning. This platform benefits from the fact that the creation and edition of games turned to the educational environment is quite simplified compared to platforms with similar objectives, making it a good option for this purpose. Moreover, its quiz games allow collaboration with the use of helps between players for completing the game activities, and competition is stimulated by giving rewards and penalties along the game and ranking players according to their performances [2]. Besides that, the fact that the creation and development of the games is performed on a web application might lead to some inconveniences when dealing with internet connectivity problems. In order to increase robustness to the platform and allowing the creation and development of games offline, as well as a greater control of personal data and resources, a Desktop Application was developed for the

new version of the Epik platform, selecting the right frameworks according to conditions regarding the needs and limitations of development.

In order to allow the platform's new version to provide games performing activities of different kinds in the same game, Interactive Puzzle activities were developed, extending the set of activities available for the games in the platform. As the main objective of this thesis, puzzle games are one of the most challenging type of games, demanding deep thought, logic and understanding in order to solve them. A whole range of themes and topics may be included in puzzle games, meaning they can be adapted to almost all subjects, allowing the development of cognitive skills, increasing visual spacial awareness and developing a deeper understanding of these themes and topics. Students must think and develop strategies, whether in collaboration with others or alone, on how to approach in achieving a goal. This process involves problem solving, analytical, memory and reasoning skills and developing solutions, according to the type of puzzle, which can later be transferred into their personal/adult life [10]. All things considered, the inclusion of interactive puzzle games might have a positive impact on the overall Epik platform experience for the users.

1.2 Framework and Context

This dissertation is part of the project 'Restructure, Flexibilize and Update the Epik Platform', which aim was to reimplement a previous work done in a previous master thesis by a student (Bruno Sampaio)[2] and extend its features, while restructuring others in order to support new features and educational activities. In this new version of the Epik platform, the creation and development of Epik games is made in a Desktop Application instead of a web application as in the previous version, and its execution is provided by a dedicated server. This change in the development provider was necessary in order to make the platform more robust and flexible, once it allows users to develop games without having the need to be online, offering a bigger control on their personal data and resources, as well as absence of failures and resultant lost of information in case of losing internet connectivity. The need to be online is only necessary for downloading the Desktop Application and activities plug-ins to be installed in the application, where the interactive puzzle activities are included, to import the generated games from the Desktop Application to the Epik Website in order to make them playable for other users or to play the Epik Games, whether it is in a Single or Multiplayer strand.

A group of five students was in charge of developing this project, having a common ground in which all collaborated and agreed, as well as independent tasks which were implemented by each student individually. Despite these tasks being accomplished individually, they still had to be coherent and cohesive with the design and functionalities of the remaining common work developed. With that in mind, follows a list with containing these tasks and their corresponding developers:

- **Scenario/Activities flow**: Graphical component to draw and improve the flow of the connection between scenarios, making it more attractive for the developers (Tiago Castanho);

- **Collaboration Mechanisms**: Development of new mechanisms of collaboration between players in order to perform Multiplayer activities (Alexandra Silva);

- **Question Activities**: Extend the range of the activities set, adding new Interactive type of Quiz activities (Daniela Santos);

- **Puzzle Activities**: Extend the range of the activities set, adding Interactive Puzzle activities (André Catela);

- **Programming Activities**: Extend the range of the activities set, adding Interactive Programming activities (André Correia);

To note that, for the three types of new activities to be developed, the scenarios flow and collaboration mechanisms allow the players to make use of the full potential of a game containing a set of these activities, which emphasizes the need to have a cohesive and coherent platform despite having independent tasks distributed among the students in charge. Furthermore, the implementation of the new activities allows the platform to provide different types of activities in a game, making it more complete and appealing for the players.

1.3 Main Objectives

As briefly mentioned in the section 1.1, the Epik Platform consists in a web-based framework dedicated to the management of didactic contents and development of educational games. Making part of the project 'Restructure, Flexibilize and Update the Epik Platform', there's a common part in which all involved students cooperated in order to keep the consistency and coherence of the platform among the different individual parts developed by them. With that said, the main common objectives in which all students participated were:

- **Interface Restructuring**: Careful thought and agreement between all students about what was reusable from the previous version of the Epik platform and what needed to be changed and re-structured for the new version;

- **Selection of Frameworks**: Before implementing the Desktop Application and dedicated server for the new version of the platform, a framework had to be selected after some research about which of the related and studied frameworks gathered the conditions needed for the development;

- **Implementation of Desktop Application**: Reuse of some aspects of the previous structure from the Epik platform's previous version in order to implement the application for the development of Epik games, including some changes needed and extra additions;
- **Implementation of Web Application**: Reuse of some aspects of the structure from the previous version of the Epik platform in order to implement the web application whose server is responsible for the execution of the Epik games, including some changes needed and extra additions;

Regarding the previous version of the Epik platform, only quizzes with short answer and multiple choice were available in the set of activities, which didn't offer much variety to the users. Having this is mind, this dissertation's main purpose was to extend the range of the activities set, by developing a puzzle editing component to be incorporated in the Epik Games development platform which allows the addiction of Interactive Puzzle activities to the set already available. The steps to achieve this main objective were :

- **Definition of Puzzle**: Specification of what is a puzzle in the platform's environment and context and what does a piece represent in each type of puzzle;
- **Selection of Puzzles**: Research of some of the most common types of puzzles and selection of the ones which could possibly be developed and integrated in the platform, taking into account their fitting in the Epik Games concept and the limitations of the selected frameworks and components for their development;
- **Selection of Frameworks**: Alongside with the selection of puzzles, the right frameworks for the implementation of the Interactive Puzzle activities and their Development Environment had to be searched and studied in order to choose the ones that gather the requirements needed for the development;
- **Inclusion of Puzzles**: Design and development of the selected puzzle types to be included in the available set of activities existent at the Epik platform;
- **Inclusion of Development Environment for the Puzzles**: Implementation of the Development Environment for each type of puzzle, included at the Desktop Application, which enables users to develop their own Interactive Puzzle activities to be included in the Epik Games;
- **Adaptation of Epik Games server for executing Puzzles Games**: Implementation of specific interface to the game execution in order to allow the web application to run games which include puzzle-type activities.

Overall, the developed Interactive Puzzle activities should motivate players to define a strategy to complete them, improving their cognitive skills. It is expected that the addition of this new type of activity contributes to the improvement of Epik as an educational platform.

1.4 Dissertation Organization

On the whole, this dissertation includes all the research and work related to it, initially giving emphasis to the importance of games in education, specifying puzzle games and comparing their characteristics, followed by a global notion of the latest version available of the Epik platform and its purpose, presented in Chapter 2. Having that in mind, on Chapter 3 the analysis and justifications about the changes to be made to the latest version of the platform for building the new version of it are discussed, including the proposed solutions for the common work related to the development of the Desktop Application and Execution Environment, followed by the new Interactive Puzzle activities inclusion in the application, as well as the proposed solution for each Development Environment according to the puzzle sub type.

Furthermore, the implementation of all the changes made, focusing more on the Interactive Puzzle activities, is described on Chapter 4, as well as the frameworks studied and selected to be able to implement the proposed solutions. For each component of the platform, its architecture and data base models are explained and detailed, including a global scheme of all the platform. Moreover, platform testing results are analysed on Chapter 5, based also on a short survey made to the users that tested the platform, and a reflection is made on the overall dissertation conclusions and future work regarding the final results on Chapter 6.

Chapter 2

Related Work

The presented chapter includes all the research and work related to the dissertation's theme, initially giving emphasis to the importance of games in education, specifying puzzle games and comparing their characteristics, followed by a global notion of the previous version available of the Epik platform and its purpose.

2.1 Games as an Educational tool

Educational games are being backed up in the Technology-Enhanced Learning domain as strategies that can lead to worthy learning outcomes. These games emerge as an option to address the learning and motivation requirements of the current generation of students. Empirical evidences support the positive effects of computer games as learning tools. The evidences indicate that games implementing pedagogical designs can strengthen and support school achievement, cognitive abilities, motivation towards learning, reflection, attention and concentration. Furthermore, the nature of educational games is very varied, from videogames to mainstream games. Also, the majority of educational games are completely virtual, allowing students to interact with virtual representations of concepts that are difficult to access in the real world. However, there are other relevant games that include the use of physical objects (with tangible interfaces or embedding sensing technologies) that address specific educational needs (e.g. manipulation of physical objects vs symbolic representations, physical interaction for particular therapies and disabilities like interactive mobile AR, etc). Moreover, other games are played in physical spaces considering real objects and their locations to facilitate contextualized learning [7].

In regard to more common seen games and their usage for learning, there are a lot of examples, whether they are real world or virtual games (or sometimes both), such as:

- **Bingo**: There are countless ways to use bingo as a learning tool. It can be used to

create games regarding what students are learning, such as phonics, vocabulary, spelling, sounds of letters, among others;

  - **Educational Bingo**: For example, the boards may have faces of important people of the history of the country on them and the clues are all standards-based facts.
  - **Maths bingo**: For example, the boards may have numbers and the clues are problems that the students have to do mentally or with scratch paper (Figure 2.1 on the left);

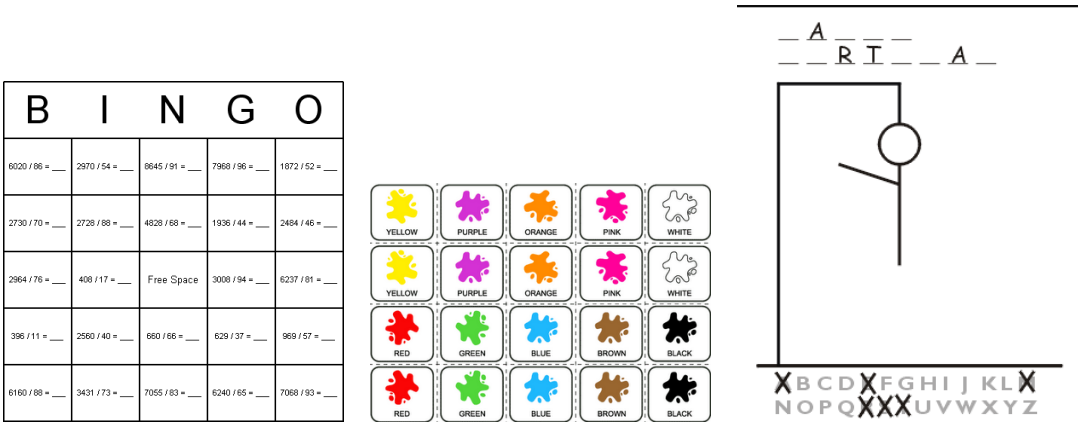

Figure 2.1: Examples of educational games [3]

- **Around the World**: All students stand up, the first student stands next to the student behind them. A card is showed and whoever is the fastest to say the correct answer moves on. For a bonus, if a student makes it all the way around the room then they get to go against the teacher;
- **Memory**: Regarding themes such as phonics, simple math expressions, spelling, and states with capitals, for example (Figure 2.1 on the middle);
- **Hangman**:Useful when introducing new topics or clues as to what the next subject of the class is going to be about (Figure 2.1 on the right);
- **Get Out of Here**: In order to get out of the classroom right before recess or lunch, students need to answer 3 questions correctly. If not, they head to the back of the line and start over;
- **Scavenger Hunts**: Despite taking some planning and time to set up, can be used for nearly every subject. Students are divided in small groups and have to search the textbooks, encyclopedias, online, and around the classroom for the clues to solve a riddle related to the subject in study;

- **Student-made Games**: A great opportunity for students to show they creativity, skills and knowledge.

However, in formal education, games are still often seen just as an unserious activity, as more of a distraction than an instructional tool, and the potentials of games for learning often stay undiscovered [6]. Teachers are very often demanded to always follow the standards and do not broadly adopt educational games in formal learning settings, which may sometimes not be the best way to keep the students attention [3]. The reasons behind the low adoption include that the available games do not often fulfill the requirements of particular educational situations, and that teachers do not have advanced technological skills to create or adapt their own educational games. Diverse research efforts are being devoted to provide easy-to-use game editors, such as e-Adventure, for example. Some other authoring tools have been developed to allow teachers to design educational games. Unfortunately, these tools can be still too complex for some instructors, hard to adapt to individual courses and require much time for development [7].

Bringing games to bear in education is not a matter of oversimplifying the content. If humans think immeasurably better as part of a network than on their own, then games are an obvious terrain in which to set minds free and let them have the freedom to make their own choices while interacting with others. The system of points, badges, rewards and leaderboards can be replicated in an educational context, to account for people's different motivations and needs for interaction or self-expression.

Further, games help students develop non-cognitive skills such as creativity, teamwork, patience, decision making and discipline to follow the rules. And those non-cognitive skills – that is, not what we know but how we behave – are far better suited to a game context than to a traditional classroom and textbook context. The traditional classroom, in many ways, stifles some of the attributes most crucial for human learning: persistence, risk taking, collaboration, problem solving [11].

2.2 Puzzles as an Educational tool

Regarding specifically puzzle-based games, they have been used to challenge the intellect for many centuries. As our awareness of the world surrounding us has grown over the decades, so too has the sophistication of the puzzles that challenge our intellect. Many puzzles have been used both as building blocks but also as educational tools for young persons. The most common examples in general public's consciousness are the puzzles known as the Rubik's Cube or the Lego building system. Each of these puzzles has challenged the intellect as well as taught the user about physical qualities of the puzzle and the world surrounding the user [12].

Because students have different learning styles, it is important to incorporate multiple teaching techniques into the classroom experience. One such technique is the use of puzzles and games in the classroom to reinforce the learning objectives and development of

cognitive skills, such as reasoning, attention, memory and problem solving. Many topics and themes are well suited for coverage in puzzle games. In basic and advanced courses, simple crossword puzzles reinforce terminology and Jeopardy-style games help students master material with short answers [13]. Moreover, experiences with spatial toys such as blocks, puzzles, and shape games have a significant influence on the development of spatial skills. Spatial skills is the capacity to understand, reason and remember the spatial relations among objects or space, which are important for success in science, technology, engineering, and mathematics fields [14].

Furthermore, puzzles are also a great educational tool to enhance and promote cooperative play. As students work together to complete a puzzle, they will discuss which place a piece should fit and the reason why, take turns, sharing and supporting each other when handling frustration, then sharing the joy of finishing the puzzle. The accomplishment of achieving a goal brings satisfaction. Overcoming the challenges involved in solving a puzzle may provide a boost to self-confidence and self-esteem as this prepares students for other challenges in life [10]. Moreover, recent studies have been made that explore a new way to integrate advanced display technology into educational puzzle activities for children with different disabilities. A free interactive mobile AR application was developed to facilitate the learning of geometry. Twenty-one elementary school children participated in an experiment. The results show that the AR system could help the school children to finish puzzle game activities independent of teacher's assistance. With the use of AR display technology, the participants demonstrated improved ability to complete puzzle game tasks when compared to the use of traditional paper-based methods. Performance data indicated that the use of AR technology could enhance learning motivation and frustration tolerance in children with special needs [15].

That said, follows a list with some examples of puzzle games used as learning tools, whether real world or virtual games (or both in some cases). The proposed games could be used in laboratory sessions, as a complement to traditional approaches. Students could also play these games as a homework assignment to reinforce the concepts learned in classroom [7]:

1. **Line puzzles**: Puzzles where numbers are presented, each one representing points that make part of an image. The student must connect the numbers in an ascending order to be able to discover the resulting image after connecting all the numbers, as shown in Figure 2.2. In this type of puzzle, a piece is considered to be each line resultant of the connection between two points, since when all the lines are connected, the puzzle is completed [16]. This puzzle essentially enhances the development of visual processing once even without finishing the puzzle it is possible to have a notion of what the final image is going to be;

2. **Sudoku**: Students need to fill a 9×9 grid with numbers so that each column, each row, and each of the nine 3×3 sub-grids that compose the grid contains all of the digits from 1 to 9. The puzzle setter provides a partially completed grid, which for a

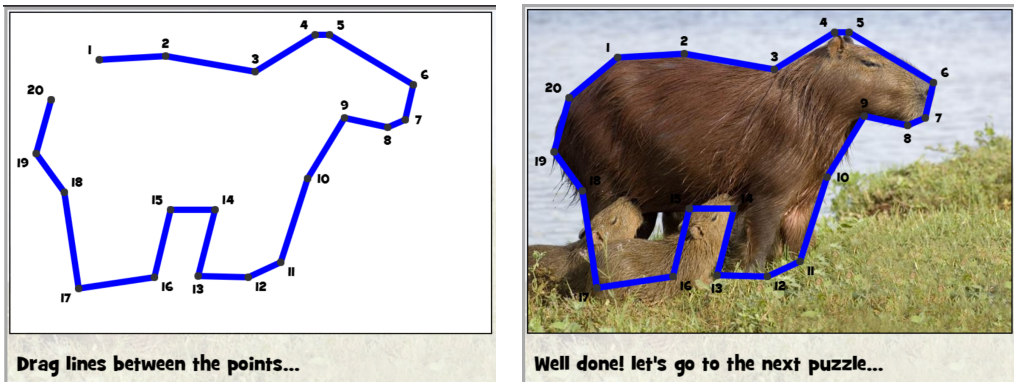

Figure 2.2: Example of a line puzzle game where all numbers must be connected in an ascending order to complete the puzzle [16].

well-posed puzzle has a single solution. In this case, a piece is considered to be each number, since filling a number in the right place contributes to complete each of the nine 3×3 sub-grids and, in a bigger view, the puzzle itself, and to find out other numbers that might not be possible to find without that piece [17]. Sudoku is a very complete game in terms of skills developed, since not only logic and reasoning needs to be used in order to verify all the options available for a square until finding the right ones, but also attention needs to be always present when searching for the correct number positions.

3. **Crossword Puzzles**: Students are given a list of clues across and down and they need to fill the correspondent number on the board with a solution to the given clue, as the Figure 2.3 suggests. For this puzzle, a piece is considered to be each letter, since the set of letters read in certain orders form the solution and the same letter can be part of more than one solution, and when all solutions are found, the puzzle is complete. In this puzzle reasoning is the most developed skill, since one needs to use letters from words that will help find new words. Also general knowledge is developed once some words may be new to the players;

4. **Jigsaw Photo Puzzles**: This type of puzzle may be of different formats. In some cases, students are given a piece of the puzzle with a question (it can be mathematical, a statement, a fill in the blanks, among other possibilities), and the different options appear in the remaining pieces of the puzzle to complete, as demonstrated in Figure 2.4. The student needs to match the piece with the question to the correct answer in order to put the piece in the correct place. In other cases, the pieces may have the answers and the questions are spread among the blank spaces of the puzzle to complete. Furthermore, we can have an entire row/column with answers for completing a fill in the blanks question given by each piece of the puzzle [16]. For this puzzle, visual processing and reasoning are the two most developed skills, since the correct position of the pieces might be discovered by solving the given

AFFIRMATIVE

ACROSS
1 Shade provider
4 Vinegary
10 Drop from a list
14 One of the Stooges
15 Thin layer
16 Pianist Peter
17 "Yes!"
19 Interstate sign
20 NBC peacock, e.g.
21 Joule fraction
22 Essayist Nin
24 Thomas Gray work
26 "Yes!"
29 Actress Kidder
31 Smidgen
32 Rock concert equipment
33 Printing goofs
36 Louver component
37 "Yes!"
41 Holster items
42 Floor leader's milieu
43 Pendulum's path
44 Cable network
46 Tall buildings
50 "Yes!"
54 Memorable dieter
55 Like some pretenses
56 Wide of the mark
58 Flow slowly
59 Use an atomizer
60 "Yes!"
63 Commedia dell'___
64 Extinguisher
65 ___-Magnon
66 Social equal
67 Barrels down the road
68 Beast of burden

DOWN
1 Insignia
2 Chicago university
3 Scant
4 Everybody
5 Made the scene
6 Manicurist's material
7 Portuguese footballer
8 Country lodge
9 Honeydew relative
10 ___ a time (singly)
11 Capital of Lower California
12 Guy from Dublin
13 Preschooler
18 Lumberjacks
23 Composer Rorem
25 Time past
27 Provo resident
28 Fitting
30 La-la lead-in
34 Analyze
35 Taoism founder Lao-___
36 Cooking vessel
37 Guaranteed to work
38 Impure
39 Granola bar morsel
40 Siouan speakers
41 Schmooze
44 FedEx rival
45 Depletes
47 Beethoven's Third
48 Tonsorial tools
49 Dictation experts
51 Modify
52 Chef's hat
53 Serviceable
57 Trough filler
59 Inset, for instance
61 "Alley ___!"
62 GRE takers

Time

Figure 2.3: Example of a crossword game where solutions must be found by discovering solving the clues and each letter maybe be part of more than one solution.

question or by guessing the final image result after having some pieces in the correct position;

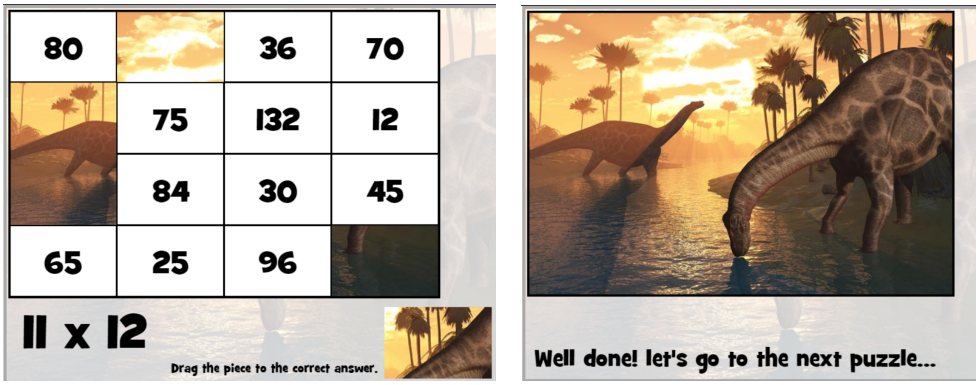

Figure 2.4: Example of a Jigsaw photo puzzle game where pieces with questions must fit in certain parts of the puzzle carrying the correct answer [16].

5. **Maths Pyramid**: A pyramid partially completed is given to the students and they need to discover the logic behind the puzzle to find the remaining values (ex. sum the value of the left and right value to obtain the upper value in the pyramid, as in Figure 2.5). A piece in this puzzle is considered to be each block that forms the pyramid, since some of the blocks are already filled with the answer and each new value found for the block will affect the solving of the rest of the puzzle [16]. The pyramid helps developing reasoning and logic, since reasoning will help to find some correct answers in order to discover the logic behind the pyramid to complete the remaining ones;

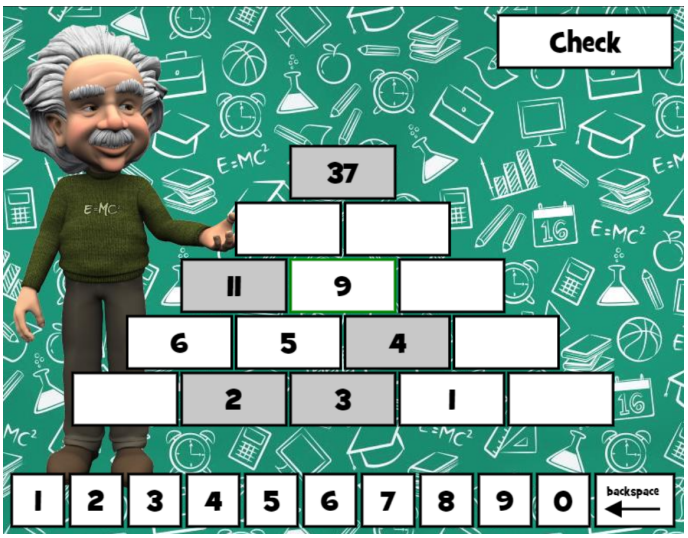

Figure 2.5: Example of a math pyramid puzzle game where each block value is found by adding the values of the two blocks below it [16].

6. **Rubik's Cube**: 3-D combination puzzle, that may have different formats, with various numbers of sides, dimensions, and stickers. In the classic cube, showed in the Figure 2.6, each of the six faces is covered by nine stickers with six solid colours: white, red, blue, orange, green, and yellow. For the puzzle to be solved, each face must be returned to have only one colour. For this puzzle, there are no representations of pieces, the cube already as all the pieces in it, but their order is mixed. The rubik's cube is a complete puzzle in terms of skills developed, where mostly visual processing and reasoning are used on the first tries to solve the cube, looking for a way to solve by looking at the colors and sequences, and after solving it a few times, the steps are memorised according to the sequences of colors visualized [18].

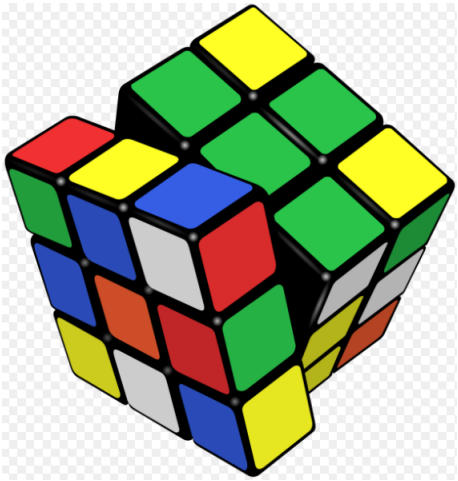

Figure 2.6: A Rubik's cube, where each face must have only one colour in order to solve the puzzle [18].

Table 2.1: Table comparing the main characteristics of each type of puzzle game in study

Puzzle Type	Formats	Themes	Difficulty
Line Puzzle	Singular	Diverse	Easy
Sudoku	Diverse	Singular	Adaptable
Crossword	Singular	Diverse	Easy
Jigsaw Puzzle	Diverse	Diverse	Adaptable
Maths Pyramid	Diverse	Singular	Adaptable
Rubik's Cube	Diverse	Singular	Hard

Table 2.2: Table comparing the range of the educational stages related and the skills involved for each type of puzzle game in study

Puzzle Type	Educational Stage	Skills Involved
Line Puzzle	Preschool	Visual Processing
Sudoku	Middle-College	Logic, Reasoning, Attention
Crossword	Elementary-College	Terminology, Reasoning
Jigsaw Puzzle	Elementary-High School	Visual Processing, Reasoning
Maths Pyramid	Elementary-College	Logic, Reasoning
Rubik's Cube	Middle School-College	Visual Processing, Memory, Reasoning

Given these examples, a conclusion must be done about the characteristics that most of these puzzle games gather, in order to have a baseline of the most adequate to be developed as Interactive Puzzle activities for the platform. Each puzzle has their own style and way to complete, showing different logics and skills developed. The type of pieces to complete the different puzzle are very disperse, or may don't even exist, as showed by the examples given. The Tables 2.1 and 2.2 gather some characteristics of the given examples to facilitate the comparison between them.

As shown in Tables 2.1 and 2.2, the displayed types of puzzles all show different characteristics, presenting different formats, themes and difficulties and stimulate different cognitive skills. The fact that in each type it's possible to change its format, theme or difficulty makes puzzles an interesting game to create, having in mind that the developer may regulate these characteristics in order to change the puzzle and it's solving, making it adaptable for a wide range of age groups and educational stages, demanding that students change their way of thinking despite being in the same type of puzzle but with different characteristics.

2.3 Frameworks for developing Interactive Games

Having gathered a notion of the importance of Technology-Enhanced Learning in increasing the motivation and learning of students, there are some frameworks available which allow the creation of interactive games. Having this in mind, a couple of examples of the most used ones is presented in the next subsections.

1. **Google Blockly**: The Google Blockly library allows to add an editor (Figure 2.7)

that represents coding concepts as interlocking blocks. It supports five different programming languages (including Python, JavaScript and PHP), or any other to user's choice. Furthermore, custom blocks may be created to connect to the application making it highly customizable and extensible. It is also compatible with all major browsers: Chrome, Firefox, Safari, Opera, and Internet Explorer [19] [20].

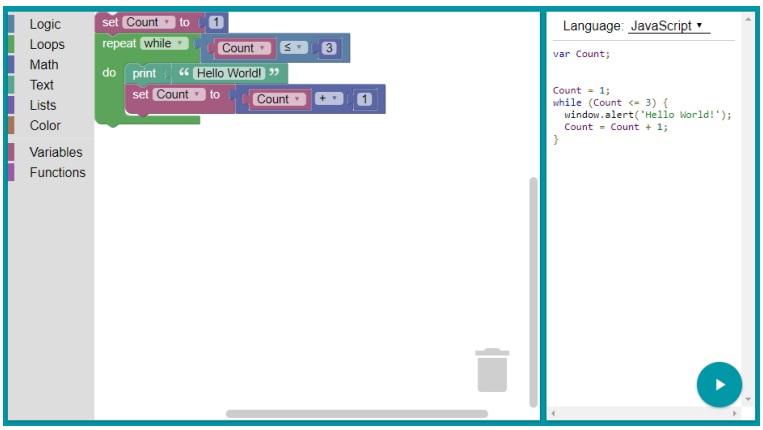

Figure 2.7: A simple example of interactive game development in Google Blockly environment [19].

2. **Scratch**: Scratch Blocks is based on the Blockly technology from Google, allowing the graphic programming of blocks. The Scratch Blocks allows the construction of programming blocks in vertical based on text, and horizontal formats based on icons (Figure 2.8). By being easy to use, it helps youths thinking in a creative way to create their games and interactive animations [21].

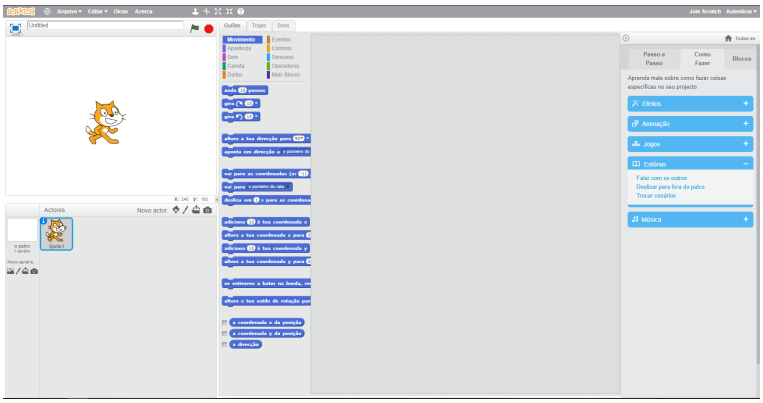

Figure 2.8: General View of Scratch Integrated Development Environment based on Google Blockly technology [21].

3. **Snap!**: Snap! is a visual, drag-and-drop programming language for kids and adults that is also a platform suitable for a serious study of computer science for high school or college students. Implemented using Javascript and running in a browser (Figure 2.9), Snap! is an extended reimplementation of Scratch which combines

drag-and-drop interface, visual metaphors for loops, conditionals, and easy animation tools from Scratch that allows building custom blocks [22].

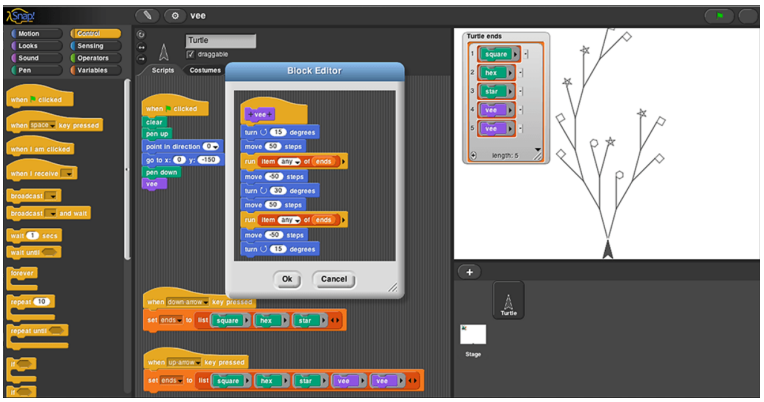

Figure 2.9: General View of Snap! Integrated Development Environment running at a web browser.

4. **Construct 2**: Construct 2 allows drag and drop objects, add behaviours to them, and make everything come alive with the list of events available in its IDE, meaning that there is no need to use programming skills for development (Figure 2.10). However, some additional complementary functionalities will require the use of python programming language. The Layout Editor provides a visual interface to design the game. It's possible to drag, rotate and resize objects, visualize effects applied, and quickly change their settings in the Properties Bar. Objects can be arranged on separate layers for enhanced organization, allowing advanced parallax and blending effects. There's also a built-in image editor to conveniently make quick edits to object's graphics. Furthermore, after developing it is possible to export to any platform [23].

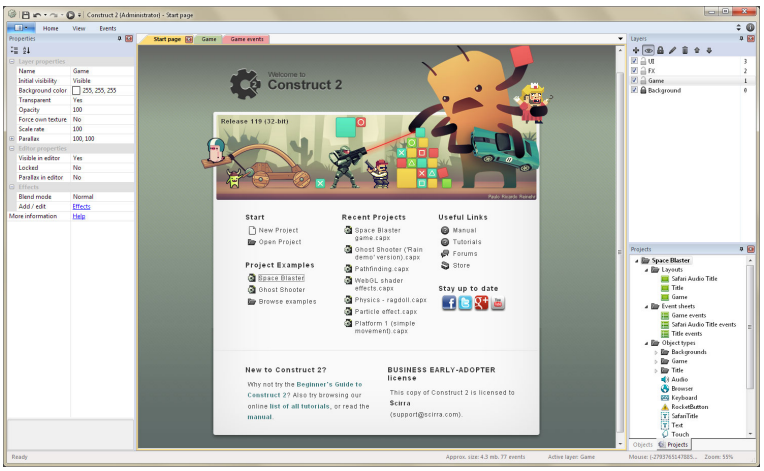

Figure 2.10: General View of Construct2 Integrated Development Environment, showing the different options for layouts, events and objects [23].

5. **Unity**: Unity is a cross-platform game engine which is primarily used to develop both three-dimensional and two-dimensional video games and simulations for computers, consoles, and mobile devices. It supports 2D and 3D graphics, drag-and-drop functionality and scripting using C#. Within 2D games, Unity allows importation of sprites and an advanced 2D world renderer to be used at its Integrated Development Environment (Figure 2.11). Additionally, there are a lot of tutorials and documentation available to help user to get familiar with the framework more easily. However, some additional futures will only be available at the payed version of the framework. After development, it is possible to export to the most common platforms [24].

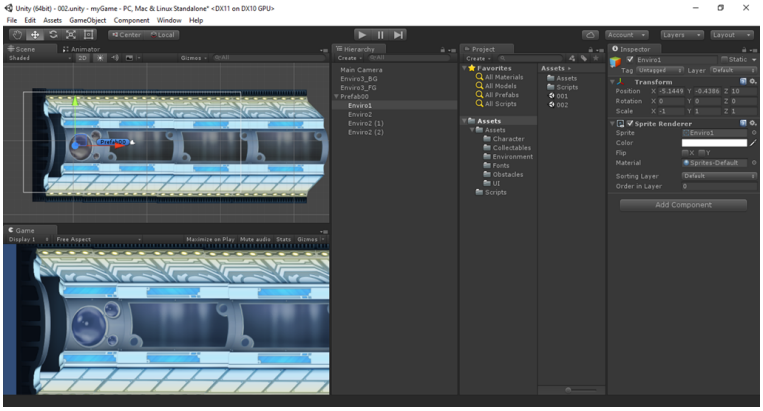

Figure 2.11: General View of Unity Integrated Development Environment, displaying a more complex but more complete interface in comparison to the previous ones [24].

2.4 Epik Platform

Taking into account the existent platforms for developing interactive games, the problem most of them show is that it is necessary for the user to have a high degree of knowledge in order to use them correctly and also some of them are not available for free. With that in mind, there is a need to have software which can enable users to develop their own educational games according to their needs, in a simple and easy way. To fulfil these requirements, the Epik platform was created.

The Epik Platform is an online application, which uses a Graphical Environment to be able to develop and manage collaborative educational games. Regarding the previous version of this platform, besides what was already demonstrated, the advantage it has to the other game creation platforms is that its quiz activities in games, despite being very similar to traditional exam quizzes, provide online distribution, immediate feedback, and some may incorporate educational resources, which allow the association of helps to the questions, making the quizzes more a learning activity and less only an evaluation activity. Furthermore, this platform has the feature of collaborative quizzes, which promotes the cooperation and interaction between users [1].

Diverse research efforts are being devoted to provide easy-to-use game editors, such as e-Adventure, Alice(2000), Squeak(1997) and GameMaker(2009). These authoring tools have been developed to allow teachers to design educational games. Unfortunately, these tools can be still too complex for some instructors, hard to adapt to individual courses and require much time for development[7].

Despite all of these frameworks integrated on Epik being able to generate and manage quizzes easily, as well as managing the students outcomes, they lack in some important features to enable the full use of quizzes as a learning activity, a problem that will be discussed later on in this dissertation.

2.4.1 Technology used by Epik

The platform is composed of two servers (Figure 2.12), one that is responsible for the development and management of games and its content, and another one responsible for the execution of the games, managing all interaction and collaboration between players, as well as the game data to be saved. [1].

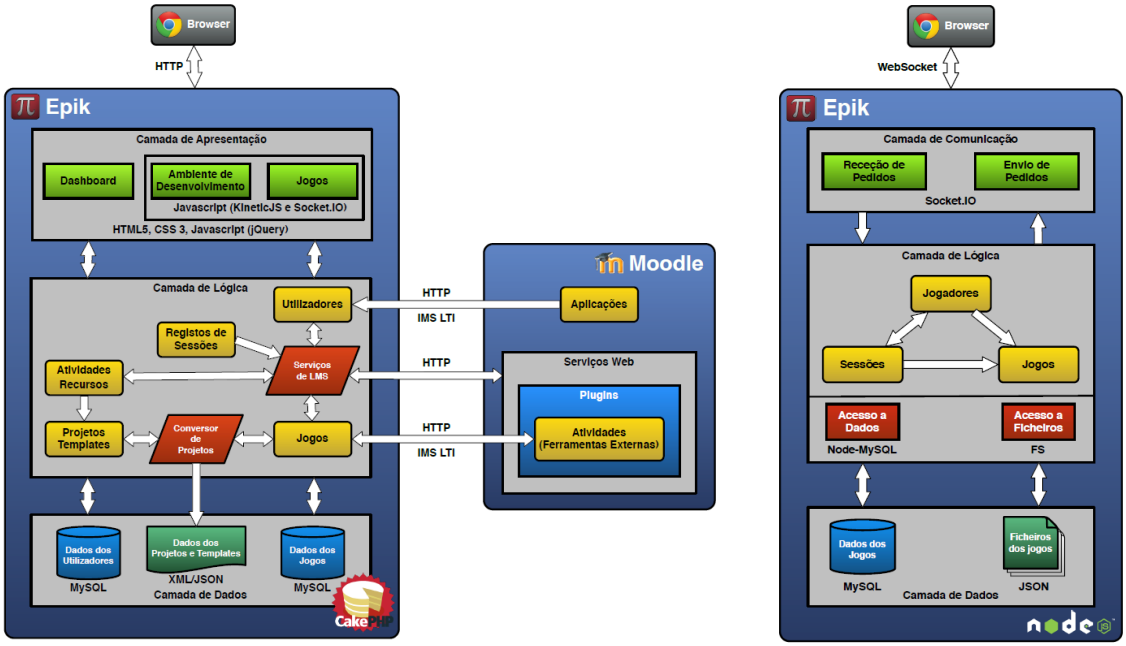

Figure 2.12: Epik Platform development and management server (on the left) and execution server (on the right) [2].

More deeply, both architectures of games development and management server (Figure 2.12 on the left) and games execution server (Figure 2.12 on the right) are presented as a three layer architecture. Regarding the games development and management server, for the presentation layer was used HTML5 and CSS3, slightly recent technologies which don't require additional features to create graphical content in a web page. This layer is responsible for generating the content to be sent to the client, based on data processed at

the logical layer. The logical layer processes all the requests of a client, using data from database or file system, accessible through the data layer. All the received requests are processed according to its component's context. The component of project conversion is responsible for converting the data of a project to other formats, allowing to convert it to the game format (XML to JSON). The data layer is composed by two MySQL data bases, one which stores data related to the users and another one for storing game data, and by the projects data in a XML format. The games execution server has a communication layer, which uses the Socket.io library, responsible for handling requests made by the clients, regarding their reception and sending the corresponding response. The remaining layers work similar to the management and development server ones. This server was implemented using NodeJS 0.8.16, a platform which allows the implementation of servers that support various types of protocols in a light, efficient and fast way, and permit the development of the server logic in the same language as the client (JavaScript). [2].

2.4.2 Game Structure and Scenarios Flow

Every Epik game is composed by a sequence of scenarios that generally follows a common flow. A scenario is intended to be the main element of a game, where all the content is displayed, including quiz activities and resources, such as images, videos, PDF files, among others. It can have an associated time to finish and a limited number of helps, both defined by the developer. In order to advance to the next scenario and progress in the game, generally players must complete successfully all the activities of a scenario within the game rules defined by the user who developed the game, such as the time to complete the scenario and helps used, for example. However, depending on the rules created by the developer, by not completing the scenario successfully the player may face only penalties and still be able to continue playing, or in other cases they lose the game.

Getting into further detail on the scenarios, there are specific scenarios, called General Scenarios, which are presented in all the games. At the initial scenario the player should fill his name and choose an avatar to be he's image throughout the game. When a quiz is collaborative, before starting, the players are positioned in a waiting room until all players have joined the game. At any instance of the game, there is always an 'Instructions' scenario available that offers a set of information concerning the actual game.

Before starting to play the initial activity of the game, or between activities in the game, generally exists a concept scenario, which may contain only text, geometric forms and/or other resources, with the objective of presenting the concepts and situate the players on the topics and rules concerning the next activities. In the activities scenario itself it is also possible to include resources such as the ones at concept scenarios, which may be used to present the activities. Both of these scenarios are included in the 'Body' scenarios.

Regarding the ending scenario, if the user is unable to complete the activities in the given time, the game ends at the 'Game Over' scenario. On the other hand, if the

user succeeds in completing all the activities throughout the game, he ends at the 'Ranking'/'Score' scenario, showing the players' score for each activity and final score for the game [1].

2.4.3 Development and Execution of Epik Games

From the previous version of Epik, the creation and development of Epik Games is made at the web application, available through browser. After registering and logging in to the platform, an empty Dashboard is presented, and the creation of a project is required in order to start developing the game, since without it is not possible to add any activities or resources created at the dashboard to the game (Figure 2.13). When creating a project, a mode most be selected, from individual to collaborative and a project template, which can be an empty one or one that is already edited. A Project is a game in development, where the users will be able to edit all the content of the game, from the scenarios design and information to the activities and resources to be included in each scenario, as well as the game flow, scores and helps (Figure 2.14). Any project, activity or resource created will be displayed in its current section at the Dashboard [2].

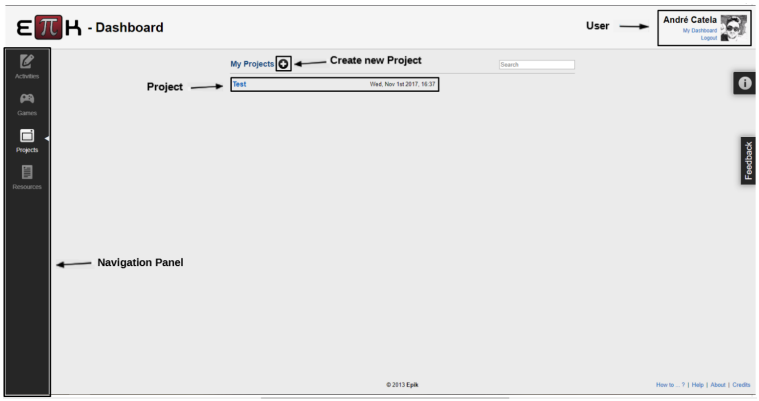

Figure 2.13: Creation of a project at Epik's Dashboard, by pressing the '+' icon near the top of the page and naming the project while at the 'Projects' tab from the Navigation Panel.

When in the project editing, a scenario must be first created from the respective toolbar option in order to add content to the game. After its creation, the scenario is selected as the current scenario, and the properties panel will allow to customize its properties, such as the name, background color, flow associated, among others. Additionally, each scenario can be decorated with forms, such as squares, circles and speak balloons, customizable at user's taste, as well as resources, like images or videos for example and all these elements that are part of the content of the scenario, including its activities, resources and shapes, are shown on the navigation panel on the left when this scenario is selected. Furthermore, more quiz activities can be inserted in the same scenario or into a new one, and the relations between scenarios have to be defined in order to indicate that

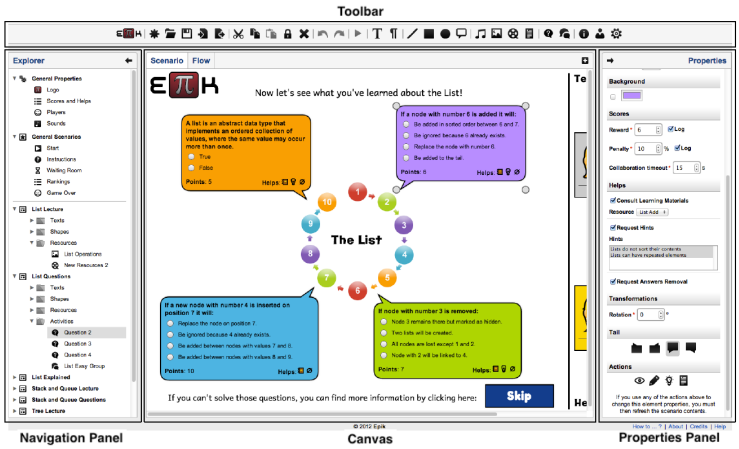

Figure 2.14: General view of Epik's Integrated Development Environment where are visible all the options from the Navigation Panel, Toolbar and Properties Panel, as well as the Canvas to preview and edit the looks and content of each scenario [1].

the players can only move to the next scenario with new activities after finishing the ones in the scenario they are in (Figure 2.15).

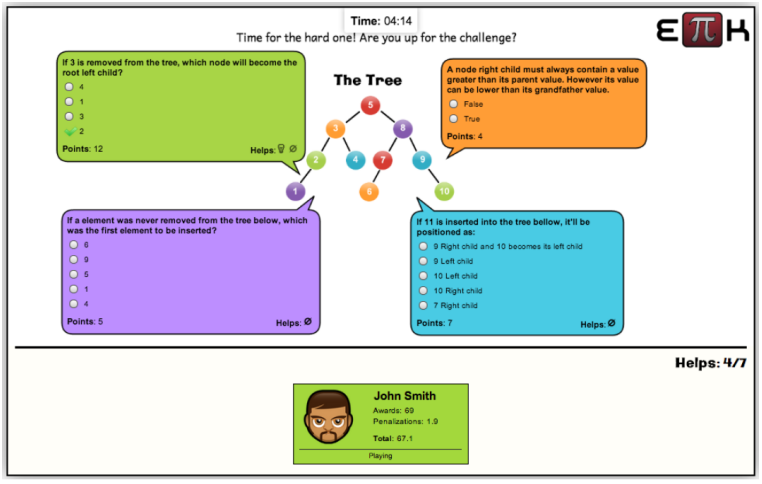

Figure 2.15: Example of a scenario in an Epik quiz game execution, where quizz activities must be answered by the player within the time limit defined for the scenario [1].

However, in order to include an activity or resource to the current project, the user has to exit the project edition environment and return to the dashboard. The inclusion of the elements on the project at the project edition environment can only be achieved after its creation at the dashboard (Figure 2.16).

With regards on activities, they are the interactive part of the game, including quiz activities of multiple choices, true or false or matching. It is possible to have groups of questions on a single scenario, from the different types of questions available. For each of the activities included in the game is possible to define a limited number of helps, score and bonus. Additionally, in a collaborative game, the time limit after receiving an help and the helps available for each question may be defined as well. The set of 'helps'

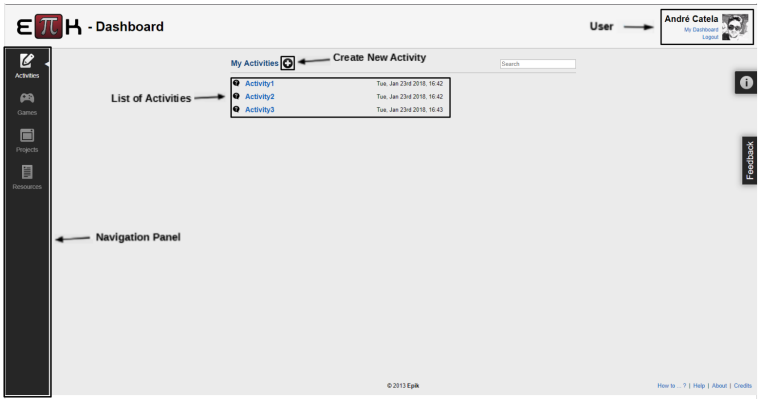

Figure 2.16: Creation of an activity at Epik's Dashboard by pressing the '+' icon near the top of the page and naming the activity while at the 'Activities' tab from the Navigation Panel.

includes didactic content(texts, slides, images or videos), hints and incorrect answers removal, also known as '50/50'. However, only team players who have already correctly answered a given question can help others. The game developer can easily configure the scores bonus and penalties values of each question, as well as the question 'helps' [1]. By using this mechanism of 'helps', Epik promotes the cooperation between members on the same team. (Figure 2.17) Moreover, each activity may be reused in other scenarios and projects and their properties may be changed according to the desired final game [2].

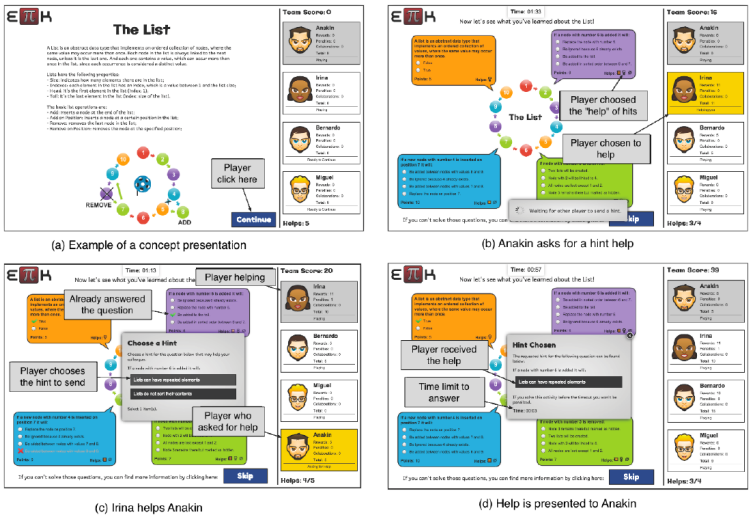

Figure 2.17: Example of collaboration regarding collaborative quiz activities, where a player that answered correctly to a question is helping another player to reach the same correct answer [1].

The Resources include content that may be added to the scenario in order to introduce a certain topic, as an help for the activities or as a decorative element. These Resources may be from different formats, including text, images,PDF, video or audio. Similar to activities, Resources can be reused in different scenarios and projects. At any time it is

possible to open an already created project or create a new project through the option on the toolbar. After configuring the whole game, some verifications must be made in order to check if the created game is valid and ready to be exported. The game must contain at least one activity and two scenarios, where an initial one must be defined, from where it is possible to reach all the scenarios in the game. The game flow may not contain any cycles between scenarios, so that the players won't get stuck in a certain sequence of scenarios. Furthermore, each scenario in game must have content, whether it is a resource, shape or activity, and at least one needs to have a transition to the Rankings Scenario, in order to allow the players to see their final score at the end of the game, even in case they don't finish it. With that in mind, all the scenarios must have a flow that will lead to the Rankings Scenario. Moreover, in case helps are defined for an activity, it must contain at least two hints. If the game is valid, while exporting the game the user may define if the game is public or not, meaning that any player with access to the game link may have access to it, even if not registered in the Epik platform. After submitting the game information, the game will be exported and stored in the server, becoming available in the games section at the dashboard. A link to the game will be provided to the students to give them access to the game.

At the end of each Epik Game, the score will be presented to the players. If its a collaborative game, the score will be the sum of all the players' scores which participated in that game session. Also an individual and team ranking will be displayed regarding all the game sessions. At the end of the game, the records of game sessions may be visualized by the user, including the player or team score for each session and the best player of all game sessions. Later on, the game creator can consult data related to each player, such as the rewards, penalties and collaboration points, number of helps used and given, bonus rewards by scenario and the points received and lost in each activity, as well as the number of attempts to solve it [2]. This is a very important feature of the Epik platform, since it allows the creator to get a feedback about the players performance, opening an opportunity to adapt the game or future games to improve the players' drawbacks.

2.5 Chapter Summary

Throughout this chapter is presented the related work of this dissertation, regarding all the theory behind and related to the thesis. This chapter first makes a brief description of the importance of games in education, making a specific reference for puzzle games and its characteristics, comparing and analysing them in order to have a notion and later on justify the chosen types of puzzles that are going to be implemented as Interactive Puzzle activities for the new version of the platform. Then connects it to the previous version of the Epik platform and its games, regarding how it works. Taking into account the content of this chapter, the changes made to some approaches in comparison to the previous version of the platform will be discussed in the next chapter.

Chapter 3

Proposed Solutions

The presented chapter includes all the analysis and justifications about the changes to be made to the previous version of the platform for building the new version of it. Initially, the proposed solutions for the common work related to the development of the Desktop Application are described, followed by the new Interactive Puzzle activities inclusion in the application, as well as the proposed solution for each Development Environment according to the type of puzzle game. Lastly, a detailed discussion about the Execution Environment of the new version of the platform is presented.

3.1 Modifications Analysis and Requirements

Taking into account what was described about the previous version of the Epik platform in the previous chapter, regarding the common part of the new Epik platform, including Desktop Application and execution server, the restructure of the interface and further new version of the platform was developed by all students involved. With reference to the previous version of the platform, the same structure and organization was kept, but there where some changes to its design and other features that required modifications in order to adapt to the inclusion of new components and functionalities. With that said, regarding the two parts of the Epik platform - development and execution of Epik Games - a set of aspects that require adjustment and subsequent solutions was elaborated.

Having that in mind, in the previous version of Epik, registration and login was required at the Epik online application before starting to create a project and developing games. Not only this takes time from users until they start developing, but also the fact that the creation and development of the games is performed on a web application might lead to some inconveniences when dealing with internet connectivity problems. In order

to increase robustness to the platform as well as a greater control of personal data and resources, the creation and development of games is now made offline, in a Desktop Application that the users must download at the Epik Website.

As a result, for this newer version, the need to login (and register if not registered yet) is only required for importing and publishing a game to be played at the Epik Website. This modification makes it faster and simpler to develop the game without wasting time on creating an account and logging in in the same session if the developer doesn't want to publish the game in that specific session. Moreover, the players won't need to register to the platform, in contrast to the previous version of the platform. In the moment of game importation, the emails of the players that are allowed to play the game will be defined. These players will receive a password, which must be inserted upon the moment they select a game in order to play it. Both players and developers of Epik Games will have a dashboard containing games they are able play or have imported, respectively, although developers might also play their own games or games other developers have assigned them to. Moreover, with the addition of new types of activities, there's a need to have plug-ins for each type, which must be downloaded from the Epik Website and installed into the Desktop Application in order to allow the creation of activities of those types. With this plug-in system, the application becomes less heavy in terms of storage and the user only downloads the plug-ins that he wants to use in his games. Moreover, for each type of activity there will be different sub-activities, which will have a specific Development Environment that will allow users to edit them. To sum up what was described regarding the structure of the new version of the Epik Platform and give a better view on the connection between both Desktop Application and Execution Environment and its users, there is a general scheme of the platform presented in Figure 3.1. Some details expressed in the scheme will be more detailed in the next sections.

Inside the game execution, the biggest change is the fact that players can now be in different scenarios, while in the previous version players would have to wait for others to finish the current scenario in order to advance. Moreover, another big change was the adaptation to the new activities inclusion, where an icon will be displayed according to the type of the activity in the scenario. When clicked, a window will open for players to complete the activity. Other minor changes regarding game execution will be discussed in section 4.3. Regarding the design, the color scheme of the new version of the platform is lighter, as will be seen in figures from the next sections, since in the previous version the color scheme was all around grey tons, making the platform slightly 'lifeless'. These were the main topics that we found a need to change initially. Other modifications will be detailed along the chapter according to the area of the platform that is being described.

Putting all together, for this new version of the platform, there are two main types of users: developers and players. The developer is a user that develops Epik games and publishes them at the Epik Website for other users to play. The players are users that will play the games developed and published by the developers. With that in mind, as a developer, the Epik Desktop Application is required to start developing Epik games. To

get this application, the developer must go to the Epik Website and download it, according to his Operative System. The application is available for Windows, Linux and MacOS. After downloading, the application will be ready to be used. However, in order to be able to create Interactive Puzzle activities and include them in the games, the developers will also need to download the respective activity plug-in and install it in the application. There are three different plug-ins added to the new version of the platform, including Interactive Puzzle activities. All things considered, the developer may start developing his games at the Desktop Application.

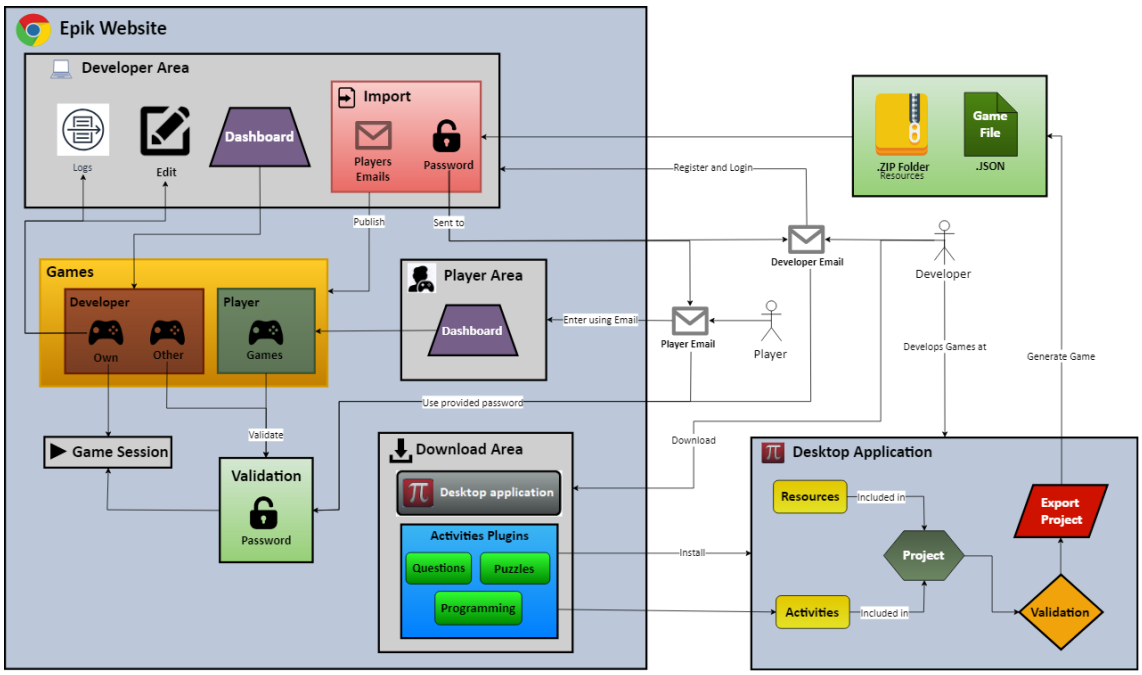

Figure 3.1: General scheme of the Epik platform from a user's perspective.

3.2 Desktop Application

The Desktop Application is where the Development Environment for the Epik Games is located. The only type of user that intervenes in the development part, achievable through the Desktop Application, is the developer. On the whole, the application is mainly composed by a Dashboard and the Project builder, which will be described in more detail in subsections 3.2.1 and 3.2.2, respectively. Furthermore, the Puzzles builder is also part of the Desktop Application, but since its a more specific component, where the different types of puzzles will be edited and each one will have its own Development Environment and functionalities, it will be described separately in greater detail in section 3.3.

3.2.1 Dashboard

Inside the Desktop Application, there is a Dashboard (Figure 3.2 on the right), where developers will be able to manage all of their games, projects, activities and resources, similar to the previous version of the platform (Figure 3.2 on the left). Initially, a default workspace is available for the developers to start developing. Each workspace contains it's own set of projects, activities, resources, and list of games, exported from the projects. Later on, new workspaces may be created and acce ssed through the dashboard, as shown in Figure 3.3.

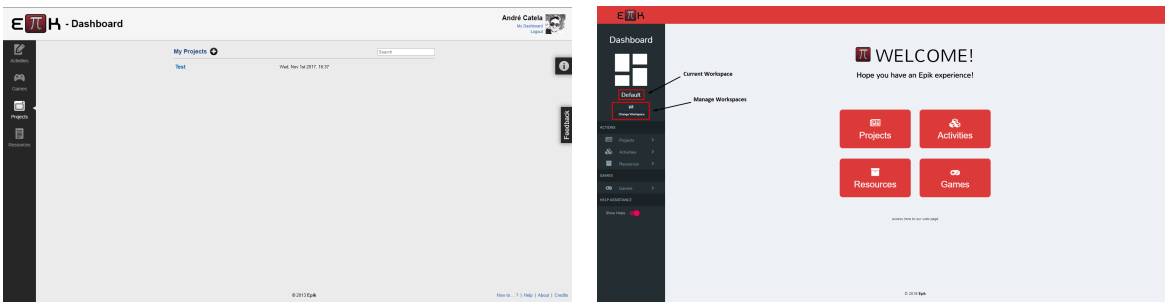

Figure 3.2: Dashboard of the Epik's previous version (on the left) and of the new version of the platform (on the right).

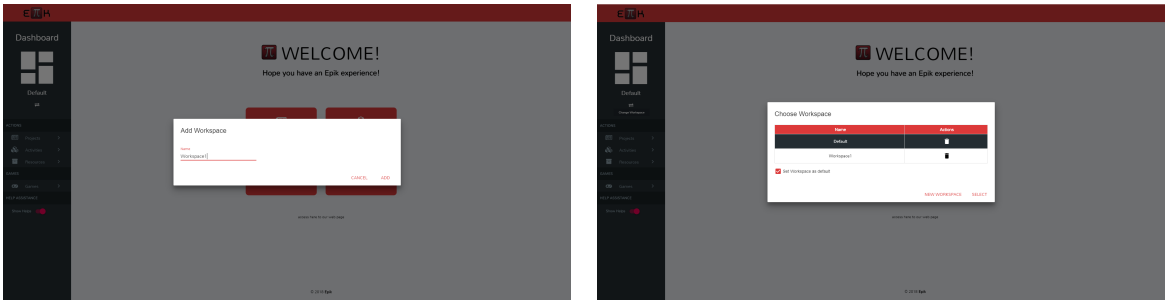

Figure 3.3: Menu to add new workspace (on the left) and list of available Workspaces to switch between (on the right).

Before going into further detail, for all the elements and functionalities of the Desktop Application there is the Epik manual (Figure 3.4). This manual basically guides the developers through the application, providing them with a tour while they are exploring it. For each topic, the manual will explain related concepts and how the developer should proceed in order to achieve the objective of a specific component of the application. After following a certain part of the tour, that part will only be available again when that area is accessed again, depending to which that part of the tour is referring to. Despite its initial utility, the Epik manual may not be of a great use to more experienced developers and therefore they may not want to be following it all the time. Having this in mind, there is the option to toggle on or off the helps from the Epik manual, at the Help Assistance

section in the Dashboard, as show in Figure 3.5. This action will only affect the Epik manual referent to the Dashboard. The Epik manual is switch on by default.

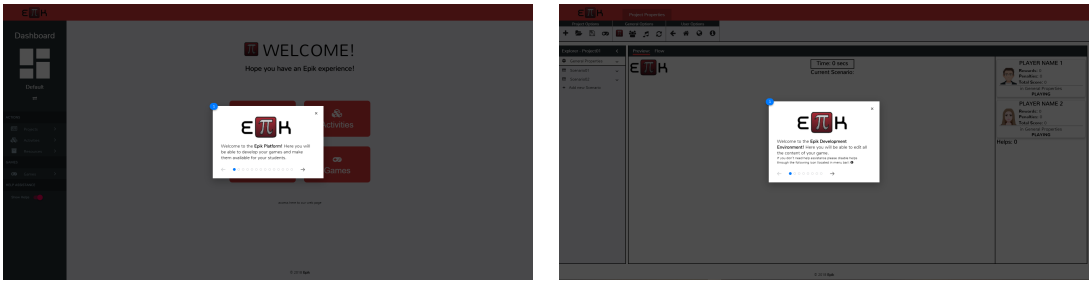

Figure 3.4: Epik Manual at the Dashboard (on the left) and at the Project Builder (on the right).

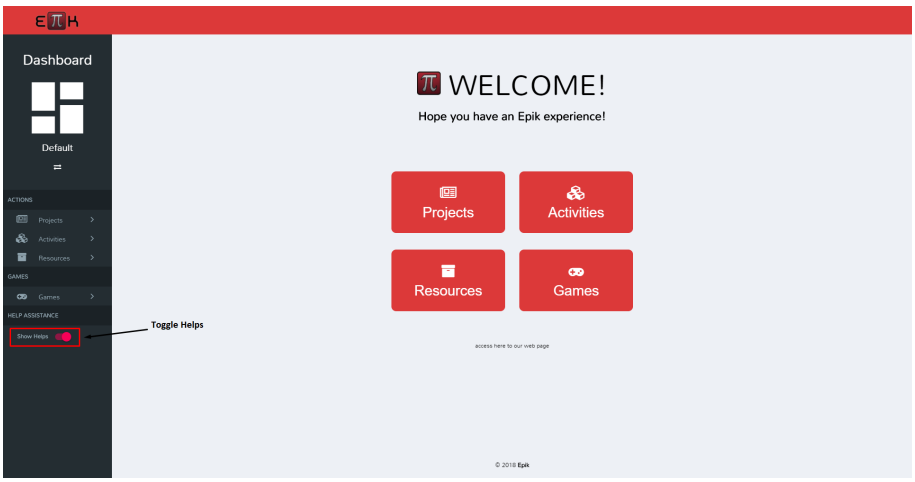

Figure 3.5: Option to toggle on/off the Epik Manual at the Dashboard

Going back to the Dashboard, there is a proper section for games, projects, activities and resources, where it is provided a table with informations about all the created items for each one of the referred elements, in that workspace (Figures 3.6, 3.7 and 3.8). At each section it is possible to add, edit or delete an item of the respective element, among other options, according to the element in sight, as shown in Figures 3.10, 3.11 and 3.12. However, if there are no activity plug-ins installed, the Activities area of the Dashboard will be empty, containing only the option to install the plug-ins to the Epik Desktop Application (Figure 3.9). When the plug-ins are installed, the developer will be able to create and edit activities according to the corresponding installed plug-ins.

Furthermore, when adding an item of any of the elements referred above, it is required to define a name for it and the option to include a description is also given. Moreover, depending on the type of element, more information must be provided. For a project, a game mode is required, between Single or MultiPlayer. An activity needs to have a type associated, such as Puzzle, for example. A resource must be an Image, PDF, Audio

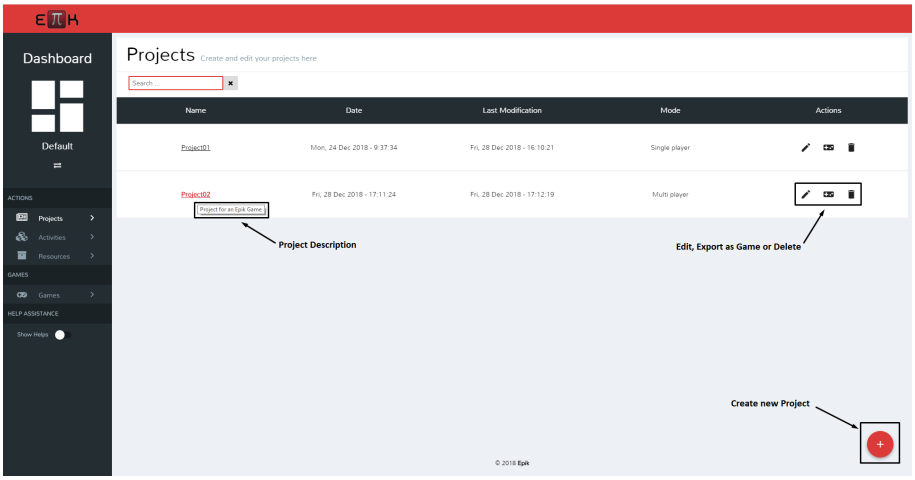

Figure 3.6: Projects Section at the Dashboard.

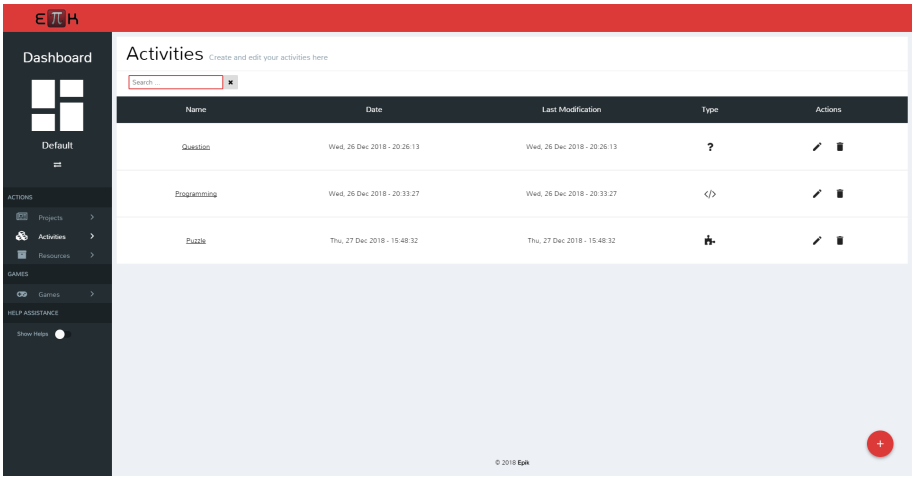

Figure 3.7: Activities Section at the Dashboard.

or Video, and include a file according to the selected type. Additionally, a search bar is provided for all the dashboard elements tables in order to filter the names of the items, helping users to find a certain item, specially when the table contains a bigger amount of rows (Figure 3.13). With that said, at the projects section, when a certain project is selected, it redirects to its Development Environment, the Project Builder, the area where the edition of the game content is made. (Figure 3.14).

3.2.2 Project Builder

Having in mind subsection 3.2.1, the Project Builder will allow the edition of all the information and properties of the game, as well as the content present in all of its scenarios and associated flow. A few changes where made regarding the project editor from the previous version of the platform (Figure 3.15).

Taking into account those changes, the new Project builder main components include:

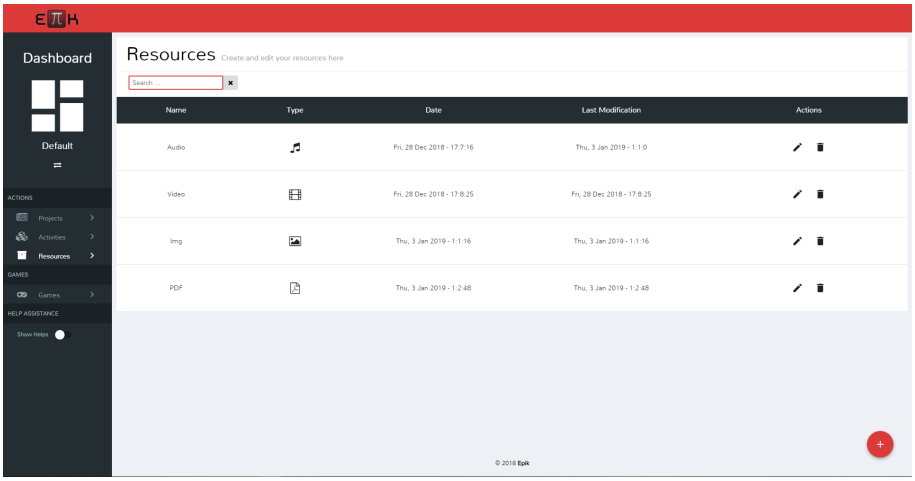

Figure 3.8: Resources Section at the Dashboard.

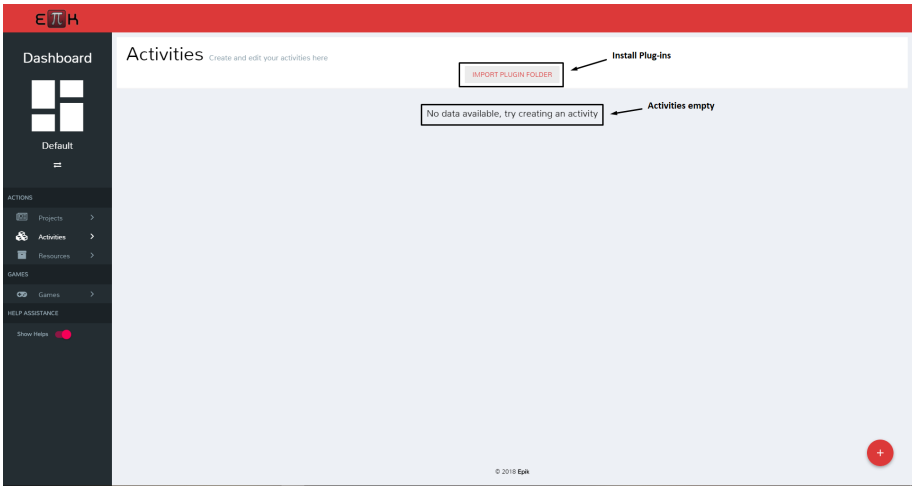

Figure 3.9: Activities Area without installed plug-ins at the Dashboard

- **Explorer**: A side bar which provides a way to navigate through scenarios and switch between them, changing the current scenario that is being edited, as well as the ability to change properties which affect all the scenarios (Figure 3.16). Furthermore, this side bar can be hidden through a toggle button, as shown in Figure 3.17, so that the preview area of the scenario is more visible and, therefore, easier to edit. With that said, this component is divided in two parts:

  - **General Properties**: Includes items that will open the respective properties tool-bar and whose modifications will affect all the scenarios created so far, or in other words, the game itself (Figure 3.18). These properties are related to the Epik Logo, Players section in the scenarios, game Sounds and Collaboration between players, and are also available at the Project properties. All the menus and functionalities regarding these properties tool-bars, will be described in more detail in the section 3.2.2.1;

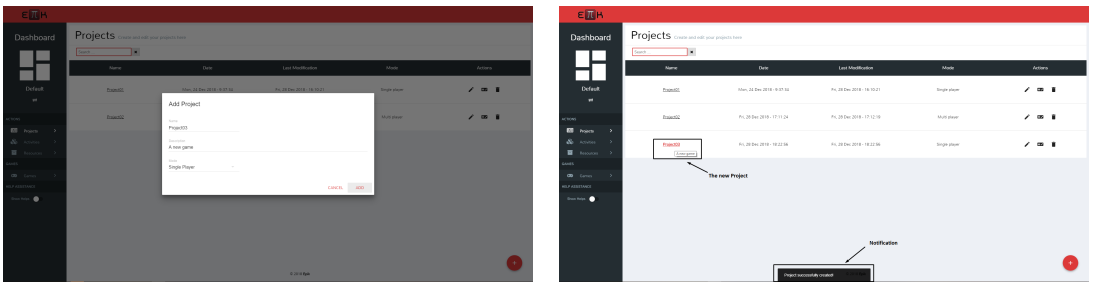

Figure 3.10: Example of the creation of a new project (on the left) and following modifications to the section table (on the right), based on Figure 3.6.

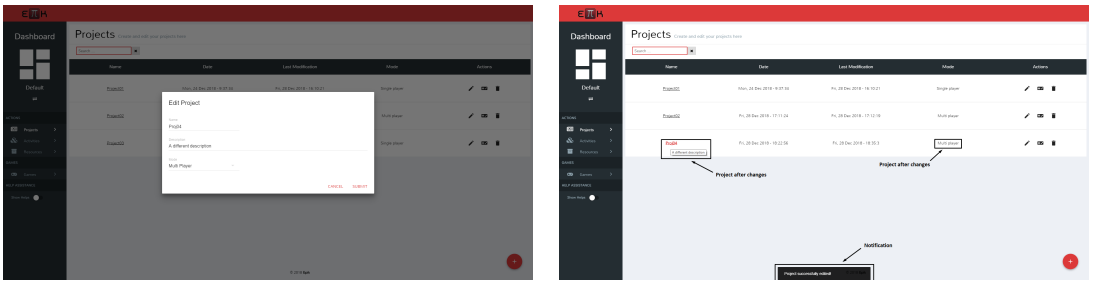

Figure 3.11: Example of the edition of a project (on the left), and changes to the section table (on the right), based on Figure 3.10.

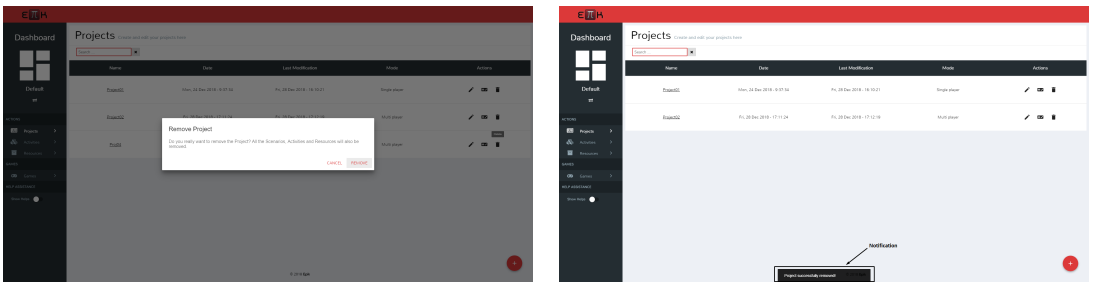

Figure 3.12: Example of the deletion of a project (on the left) and following modifications to the section table (on the right), based on Figure 3.11.

– **Scenarios List**: A list of all the scenarios created for the current project. It allows to edit and delete each one of the scenarios. On selecting a scenario from the list, this will become the current scenario and the Scenario Preview Area will show its content. Moreover, each scenario will have a list of its Resources and Activities at the Explorer, in order to make it easier to identify and open specific properties tool-bar to edit each one of them without needing to search for them in the Scenario Preview. Additionally, there is an option to add a new Scenario located below the Scenarios List, making it easy and intuitive to include new scenarios (Figure 3.19) ;

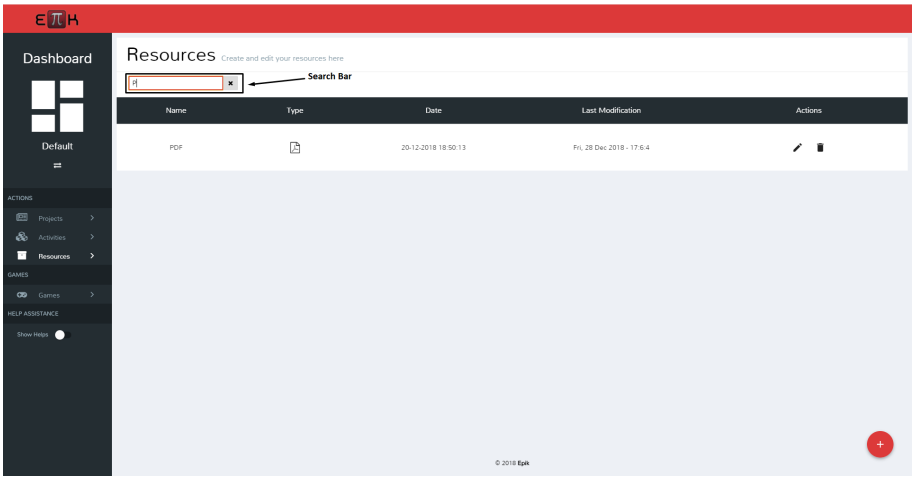

Figure 3.13: Search Bar at the Resources Section of the Dashboard. Among the elements from Figure 3.8, the search bar is filtering the ones whose names contain the letter 'P'.

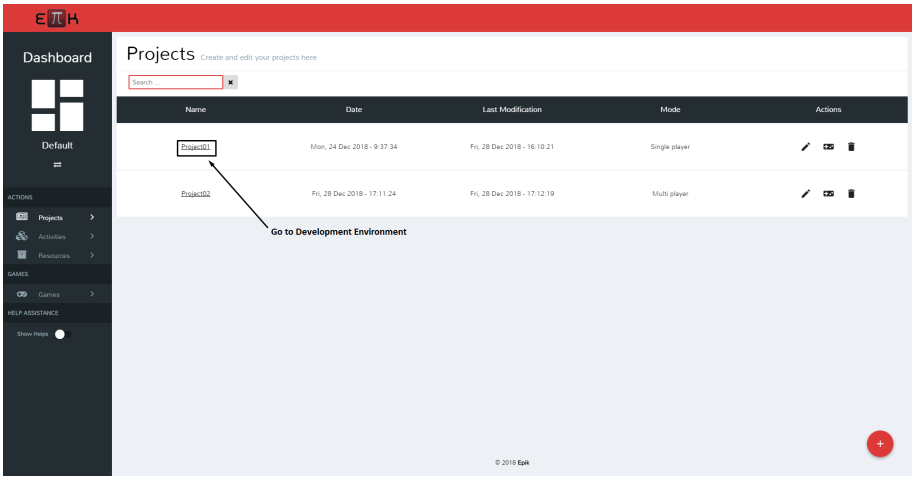

Figure 3.14: Entering the Development Environment of a project, the Project builder, by selecting a project from the list at the Dashboard.

- **Scenario Preview**: An area which shows the content of the current scenario. The name and the time to complete the scenario is also shown at the top. Changes made to the properties of the elements included in the scenario, namely resources and activities, are visible in this area. Despite these changes being done at the properties tool-bars, made available by the selection at the Explorer side bar, it is also possible to move and change the size of these elements at the scenario preview area. Selecting the element in the scenario preview instead of in the Explorer will also grant access to its properties tool-bar. Furthermore, each time an element is selected, whether in the Explorer or in the scenario preview, a shadow appears around it in the scenario preview, indicating that the element is selected, making it easier to identify it. The Players Section is also part of this area. (Figure 3.20).

  - **Players Section**: Area in the scenario where it is shown information about the

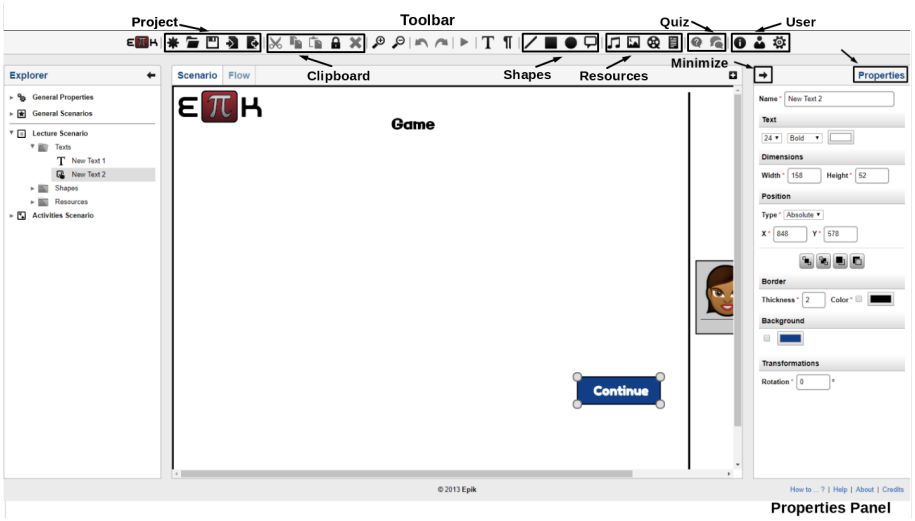

Figure 3.15: General view of the Development Environment from the previous version of the Epik platform, including the toolbar and properties bar.

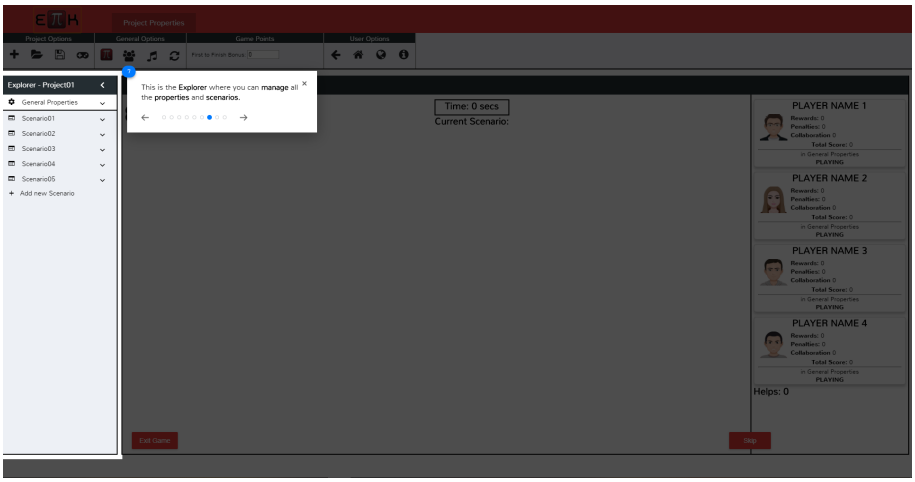

Figure 3.16: Explorer sidebar at the Development Environment with Epik manual on, containing two scenarios on the list.

players currently in game. For each player there is a player card containing information about his progress in the game. Moreover, the card shows the name, rewards, penalties and total score of the player, as well as the current scenario he is in and his state. The player state refers to whether he is currently playing, as well as if he has lost, gave up, finished or disconnected from the game. More on this topic will be discussed later in this chapter. Furthermore, helps used in each scenario will be shown at this area (Figure 3.21);

- **Flow Preview**: The area where it is possible to visualize the current flow of the game. The flow of the game is the set of paths between scenarios to which the player can go until he reaches the end of the game. An interactive component allows to move, zoom in and out the scenarios and respective transitions for a better visualization of

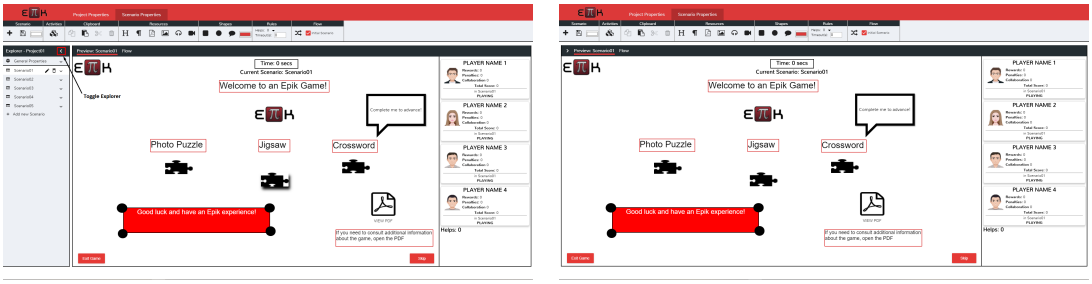

Figure 3.17: Explorer sidebar toggled on (on the left) and off (on the right).

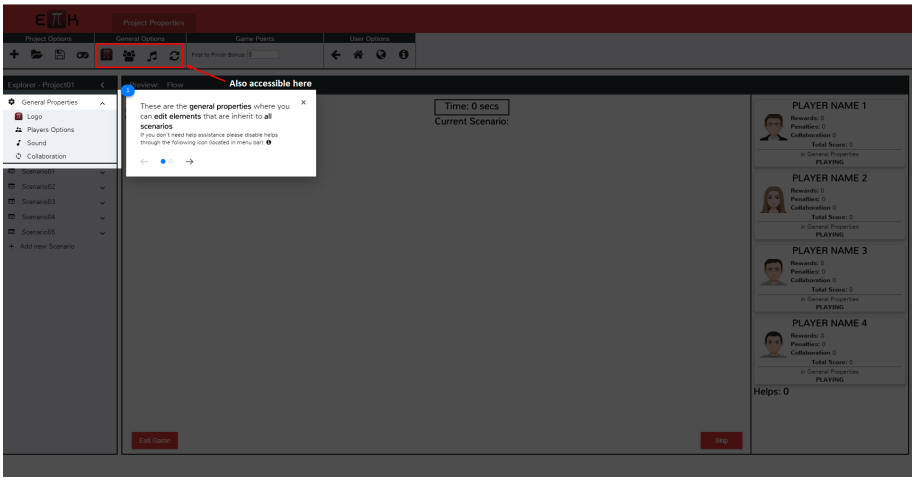

Figure 3.18: General Properties at the Explorer sidebar, also accessible at the Project Properties.

the flow graph. Each transition between scenarios has a respective color according to the action that triggers the scenario change. Consequently, from a scenario it is possible to follow different flows, meaning that the main flow of the game may change according to the performance and actions taken by the players during the game. Therefore, a list with the colors and correspondent actions is also shown in this area. Besides, these transitions between the scenarios and other scenario related properties, which will be described later in this section, can be edited at the scenario properties.

- **Properties Tool-bar**: Provides a way to edit the properties of a specific component or element of the game. The project properties tool-bar is always available at the environment's header, since it mainly contains general options which are used more often and don't require the creation of any component in order to be used. The remaining tool-bars are more specific and not as frequently accessed, so when the respective element or component is not being edited or even created, these tool-bars are hidden, keeping the environment clean and less confusing (Figure 3.22).

  Regarding the previous version of the platform, the toolbar was located at the top

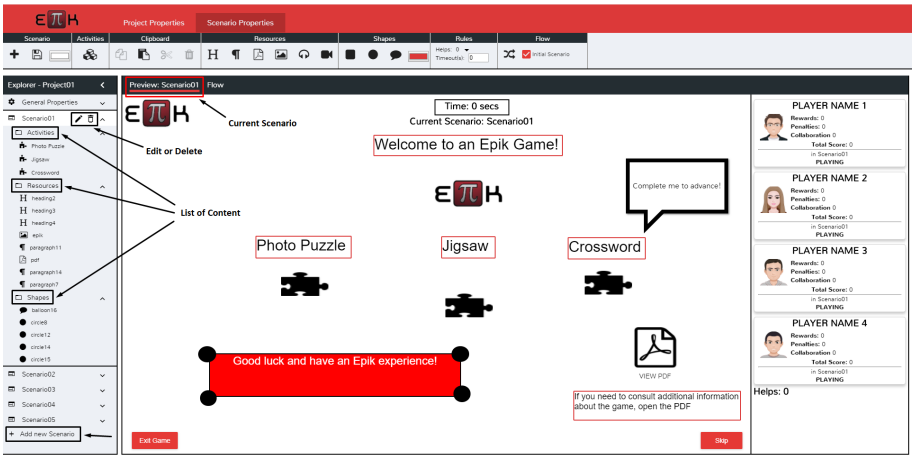

Figure 3.19: Scenarios List at the Explorer sidebar.

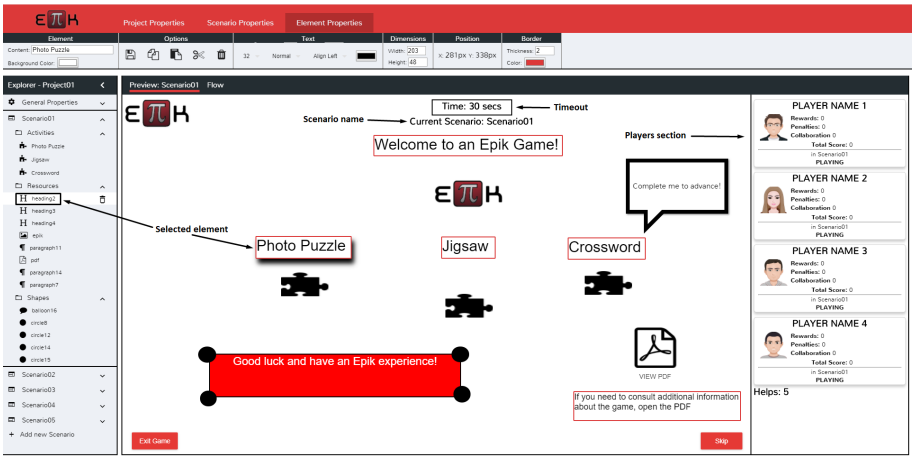

Figure 3.20: Scenario Preview at the Development Environment.

of the Project editing environment, and despite having its icons divided according to their functionality, they were too many and were displayed in a line, as shown in Figure 3.15, which would confuse some users because of the way they were displayed. Additionally, the properties panel on the right of the project editor would take up a lot of space, cutting a part of the visible area for development. Despite the possibility to minimize it, during the edition it was useful to have it open, so the developer would feel the need to be always minimizing the properties panel whenever he wanted to move objects around the scenario. With this new Properties Tool-bar, were both toolbar and properties panel are merged and placed at the top of the page, the icons and, if existent, corresponding properties are separated in tabs according to the functionalities and areas they affect, allowing to open and switch between them. The new display makes the toolbar functionalities more intuitive for the developers and creates more space for editing the project.

To summarize, the interaction with these different components of the Project Builder

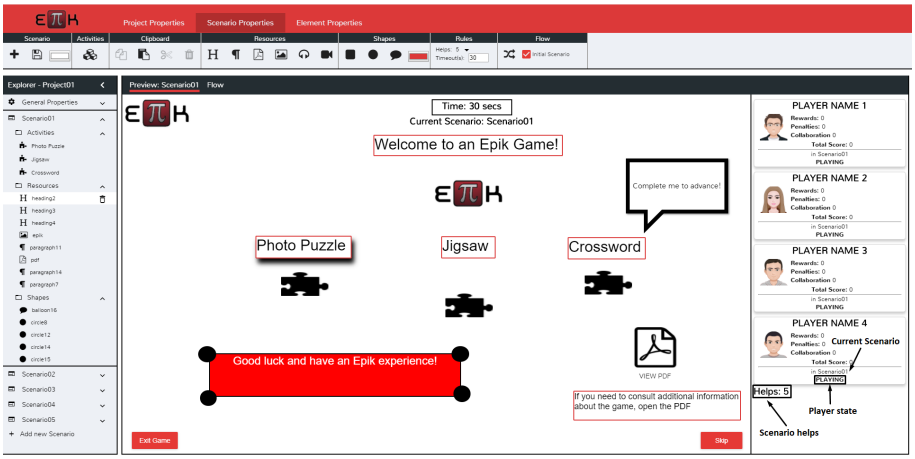

Figure 3.21: Players Section at the Development Environment.

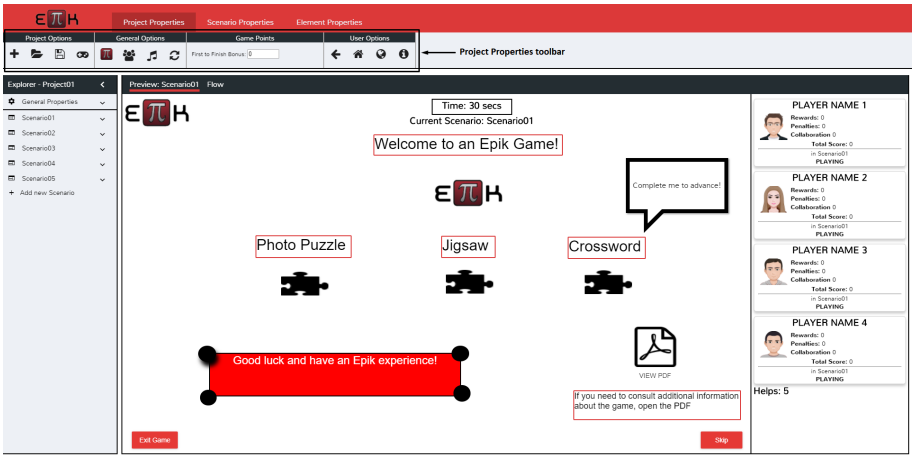

Figure 3.22: Properties Toolbar at the Development Environment.

will allow to develop the game and reach the final product. As a result, the next section will explain in greater detail the components menus and functionalities and their contribution to the changes in the game content.

3.2.2.1 Menus and Functionalities

With regard to the menus and functionalities of the Project Builder components described in the previous section, there are some properties which can be edited at the toolbar of each game component:

- **Project Properties**: Allows the creation of a new project, as in the dashboard (Figure 3.10) which will appear on a list when the option to open projects is selected,

together with other projects created in the current workspace. Moreover, this list actions provides a way to delete any of the projects or select them to be opened, switching the current project being edited at the environment for the selected project (Figure 3.23). Furthermore, the Project Options menu also allows to save the changes made to the current project and to export it as a playable game, once it is valid. The save option can be found in many properties toolbars of the application, so that it is always available without needing to switch between tabs. Game validation will be a topic discussed in section 3.2.3.

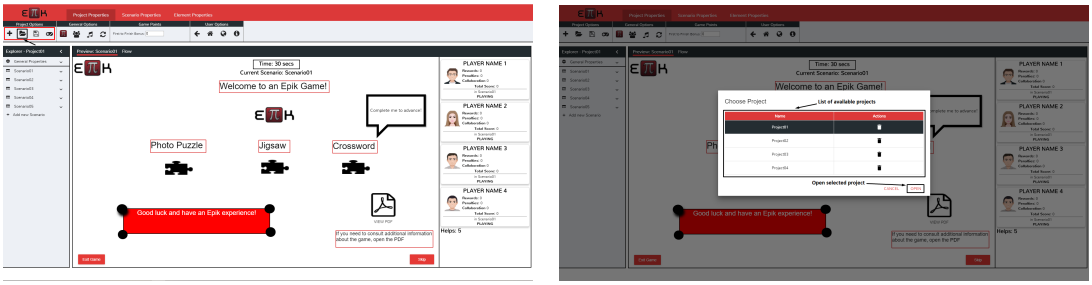

Figure 3.23: Open Project option at the Project Properties toolbar (on the left) and list of available projects from the current workspace to which the developer can switch to, after clicking the referred option (on the right).

Regarding General Options, displayed in Figure 3.18 this menu adds the same functionalities as the General Properties at the Explorer, giving the developers another way to reach these functionalities when they have the Explorer side bar toggled out. With that being said, each option will open the respective properties tool-bar, which can be:

- **Logo Properties**: Epik logo position in scenarios may be changed to the top-left or top-right corner. Top-left corner is the default position defined for the Epik Logo in the scenarios;

- **Player properties**: Affects the Players Section in the scenarios. The developer can determine the number of players of the game, if its mode is defined as Multiplayer, from 2 to 4 players (Figure 3.24). As a Single Player game, the number of players is one. The background color of the Players section can be customized, as well as its orientation on the scenario, which can be changed from the right, which is its default position, to the bottom (Figure 3.25). The thickness and color of the bar which divides this section from the rest of the scenario, regardless of its position, can also be customized. Moreover, the background colors of the Player cards is also editable, as well as the display for the rewards, penalties and total score at Player cards and the helps at the Players section, according to the theme of the game (Figure 3.26);

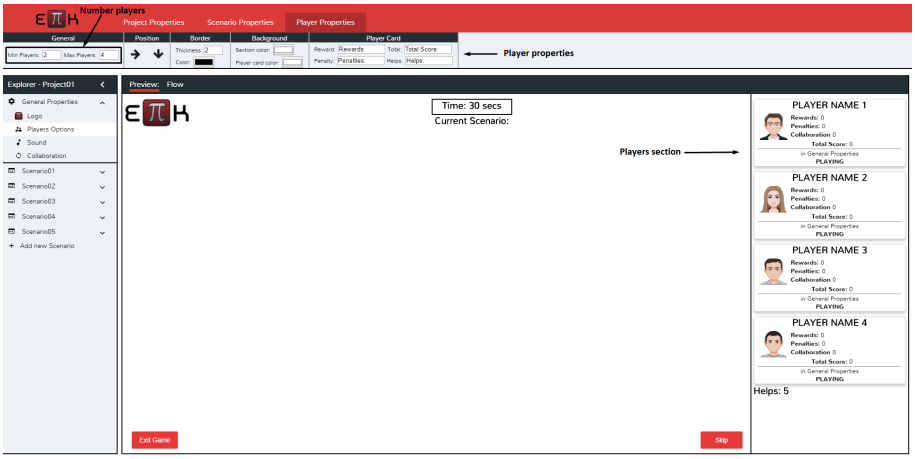

Figure 3.24: Number of Players at the Player Properties Toolbar in a Multiplayer Game.

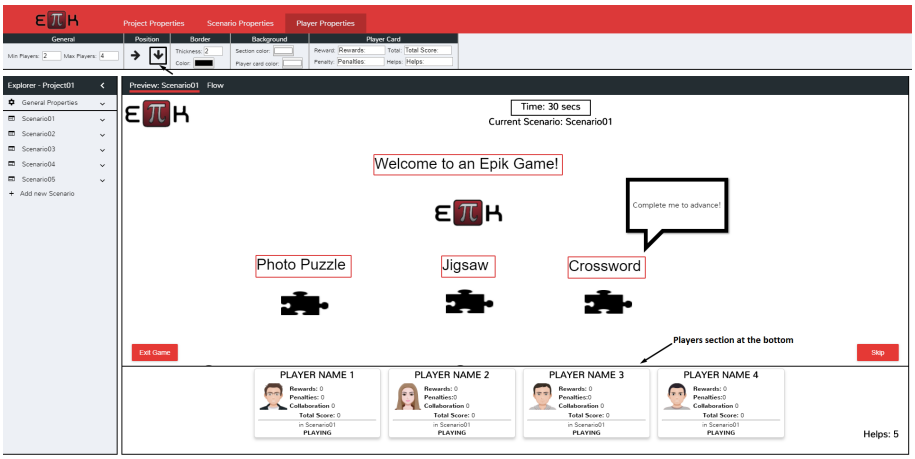

Figure 3.25: Player Section located at the bottom of the Scenario.

– **Sound properties**: Customization of the game sounds, more specifically, the background music, correct and incorrect answer sound and audio when a player asks for help. Default sounds are already provided for each one of these cases, despite the option to upload custom sounds to the list of audio resources of the current workspace and assign them according to the four options referred previously. After confirming the changes, the assigned sounds will be available to listen at the audio player presented in the toolbar. At any time the default sounds can be restored (Figure 3.28);

– **Collaboration properties**: Features related to the type of collaboration between players along the game may be enabled/disabled. One type of collaboration are the notifications, where the game will show a notification that will be visible for all the players each time a player finishes the game, asks for help, ends an activity or changes scenario. Other type of collaboration is through the

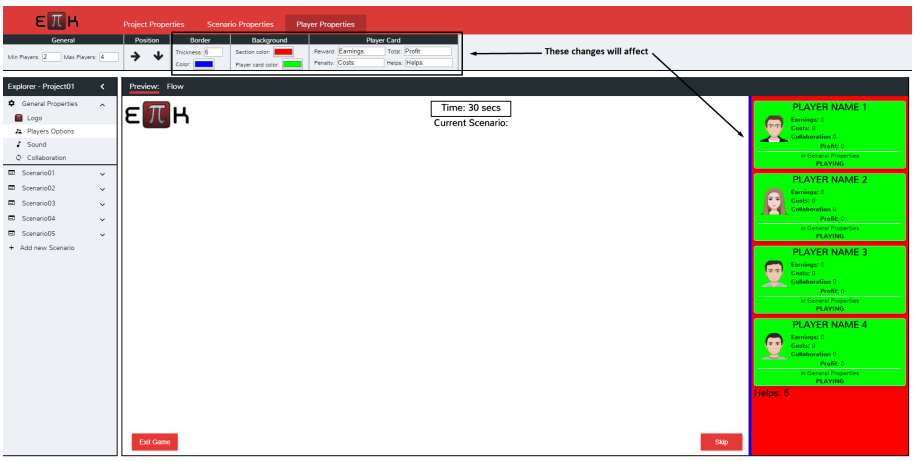

Figure 3.26: Player Properties changes at the Player Properties Toolbar and consequent visual differences at the Player Section.

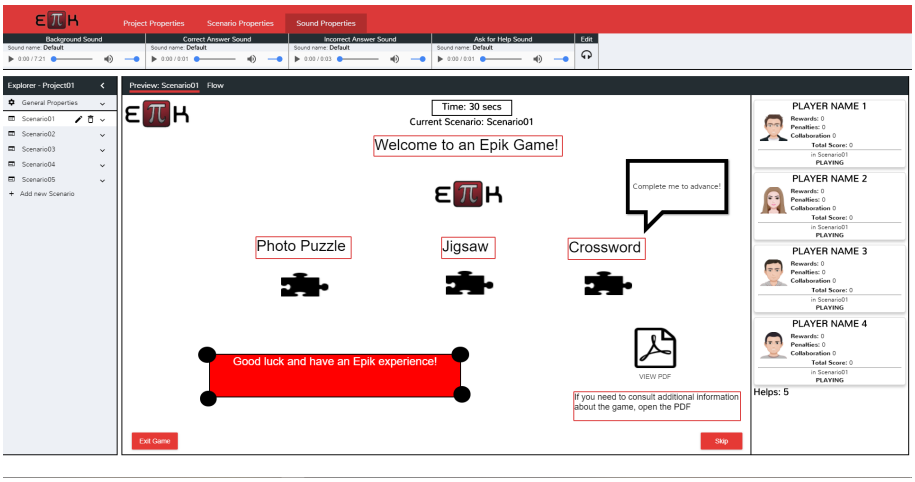

Figure 3.27: Sound Properties Toolbar.

chat, allowing users to communicate with each other through text messages along the game. Two types of chatting are available: free chatting or controlled chatting. With free chatting, players may send any text messages they want to other players, whereas with controlled chatting players may only send default messages created by the developer. A question that may arise with the introduction of the chat feature is the influence it may have in the helps for the activities in a scenario. With a controlled chat the developer can manage what kind of interaction the players may have with each other throughout the game, meaning that he can decide if the players are able to help each other through the chat. Despite this fact, these helps won't count as an help used in the scenario. On the other hand, with a free chat the developer has no control over the messages exchanged between players. Besides, in both situations the decision to help lies on the players, since they will choose which messages to

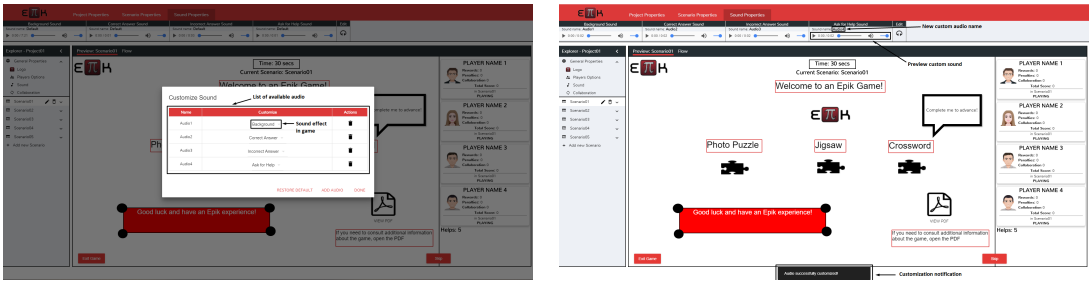

Figure 3.28: List of Available Sounds to customize (on the left) and Sound Properties Toolbar after custom changes made (on the right).

send. The last type of collaboration are the moods, which are basically emotes that players may use throughout the game to react or express their emotions according to the game situation.

Returning to the project properties, bonus points to the player who finishes first the game can also be defined. These extra points will be added to the final score of the player in the game. Regarding navigation, developers can go back to the projects section at the dashboard, where they were before entering in the Project Builder, or go back to the main page of the dashboard at the 'Home' option. Additionally, the developers have access to the Epik Website, where later on they can import a game exported from the project being edited, and to a switch to enable or disable the help assistance from the Epik manual at the Project builder (Figure 3.29).

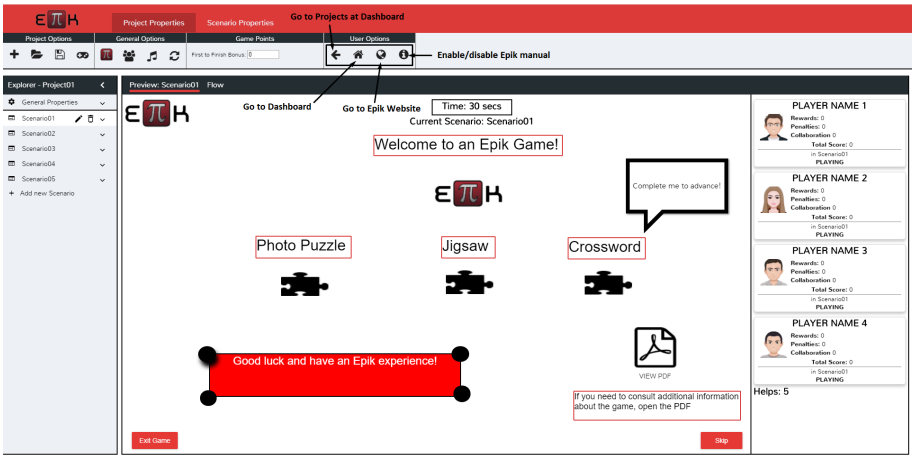

Figure 3.29: User Options at the Project Properties Toolbar.

- **Scenario properties**: Assuming that the project being edited already has a scenario created, when this scenario is selected at the Explorer, its properties tool-bar is displayed. From the previous version of the Epik platform, most of the game structure and scenarios flow was reused with a few changes. The General Scenarios from the new version of the platform, which are present in all Epik Games, include:

- **Start Scenario**: The first scenario players see before initiating the game. The game instructions are available to read, in order to know more about it before starting, or in case the players wants he may exit the game;
- **Instructions Scenario**: Scenario which contains all the game rules and main concept explanations;
- **Waiting Room**: At this Scenario, players may choose their name and avatar to be shown throughout the game. In the previous version of the platform, in case the game mode was Multiplayer, it would only start when all the players participating in the game were ready. In this new version, there is a minimum and maximum number of players in a game, defined by the developer. From the moment a player is ready, a countdown will start. If the maximum number of players is achieved before the end of the countdown, the game will start. If not, the game will wait until the end of the countdown. If in the end of the countdown the minimum number of players is achieved, the game will start. If not, the game does not start and all the players that were ready will have to start the countdown again and wait for at least the minimum number of players to be achieved. This new concept was inserted to avoid that if there is a problem with one player and he isn't ready to play the game, the remaining players won't have to leave. The player that had the issue may be later inserted in another game session with other players;
- **Finishers room**: A new scenario introduced for the MultiPlayer games in this new version of the platform. When a player finishes the game, he will be redirected to this scenario. Here he will be able to see his final score and statistics in the game, but most wait for the remaining players to finish the game, whether they complete it, loose it or disconnect, in order to see the final rankings.
- **Scores Scenario**: A new scenario introduced for Single and Multiplayer games, which contains the final scores and statistics of the players that participated in that specific game session;
- **Rankings Scenario**: A Scenario inserted in MultiPlayer games which contains the final rankings of all the players that participated in the game session;
- **Game Over Scenario**: The Scenario to where the players are redirected in case they are not able to finish the game, according to the game rules defined by the developer.

Putting all together, the game generally follows a common flow, where the Start Scenario is always the first scenario before starting the actual game, from where the users may check the Instructions or go to the waiting room and start the game itself. The Scores and Game Over Scenario, however, will appear in different occasions, according to the decisions and options taken by the players and by the flow defined

by the game developer. More on this subject of the game flow will be discussed in the next sections of this chapter. Regarding some interactions with these general scenarios during the game, they will be more detailed in section 3.4.2.

Furthermore, in the previous version of the platform, there were concept scenarios. These scenarios would contain only text, geometric forms and/or other resources and their objective was to, before starting to play the game or between scenarios, present the concepts and situate the players on the topics and rules concerning the content of the next scenarios. In this new version of the platform, the idea of concept scenario was removed, since the elements that bring the concept and rules can be included in the same scenario. Hence, there is no need to have an extra scenario just to describe concepts [1].

Returning to the scenario properties, represented in Figure 3.30 the changes made at this tool-bar will only affect the current scenario, except for the save option, which will save all the project changes, and the create new scenario option. This last option has the same functionality as the add new scenario option below the Scenarios List at the Explorer (Figure 3.19), providing an alternative to reach this functionality when the Explorer side bar is toggled out.

The Activities menu contains a list of activities available in the current workspace. When selecting an activity from the list, it is possible to delete or add them to the current scenario. When added to the scenario, an icon will appear indicating the type of the activity added. Moreover, there is the option to add a new activity to the workspace below the activities list, which has the same functionality as the add activity from the dashboard (Figure 3.31).

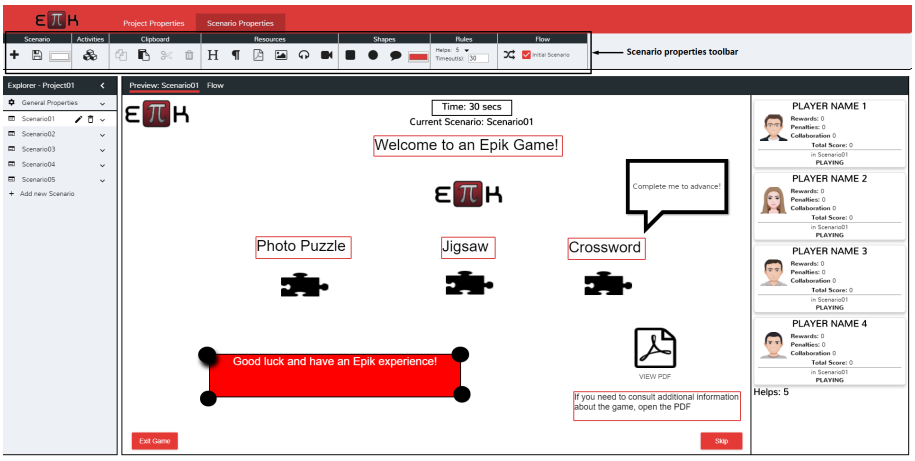

Figure 3.30: Scenario Properties Toolbar at the Project Builder.

Besides the possibility to change the background color of the current scenario, there is a Clipboard, which allows to cut, copy, paste or delete any element of the scenario. This menu is also included in the Element and Activity properties toolbar, which

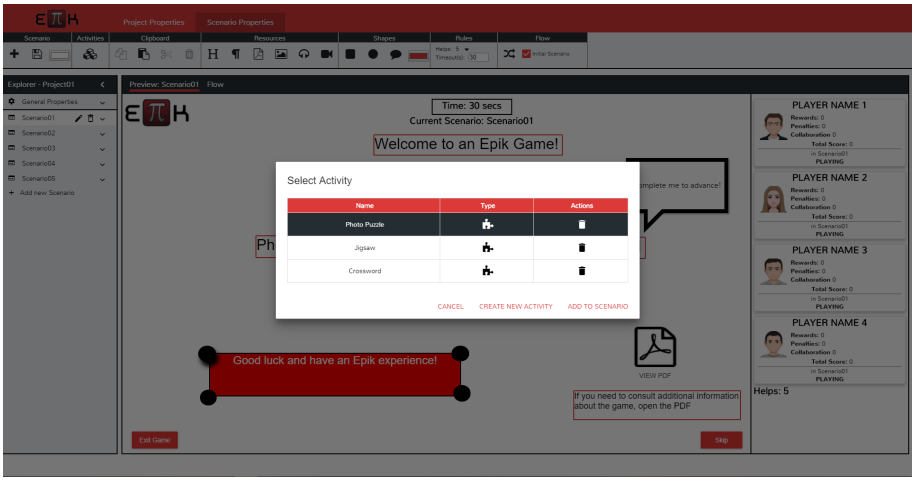

Figure 3.31: List of Available Activities in the current Workspace.

will appear each time an element or activity is selected, meaning that besides the Paste feature, all the others would not be used at the Scenario Properties Tool-bar, so they are disabled to avoid confusion. Moreover, it is possible to cut or copy an element from a scenario and paste it on another scenario from the same project or from different projects.

The Resources menu is where all the images, PDF, audio and video files are located. For each of them there is a list of the respective files already present in the current workspace, which can be selected to be deleted or to be included in the current scenario, where they can be used to introduce a certain topic, to support some information about the scenario content or just as a decorative element. There is also the possibility to add new files. When uploading new files to be added to the resources list, the upload takes into account the format of the file. For example, if the upload is for an image file, the platform only gives permission to select local files that are from the format .jpg, .png, .GIF, among other common image formats. Furthermore, Headings and Paragraphs containing text can be created to help support the scenario content. Similar to this section, the Shapes menu allows to include geometric shapes in the scenarios. Going back to the creation and development of Epik Games from the previous version of the platform, it was only possible to create a new quiz activity or resource at the dashboard, meaning that when in the project editing, it was required to already have a created activity or resource, otherwise the user was required to go back and could only include the activity or resource in a scenario of the project he was editing if they were created beforehand, which was not very practical. In the new version of the platform the creation of the activities and resources, as already explained, can be done either at the Dashboard or at the Project editing environment Moreover, The number of helps and timeout for the current scenario can be defined at the Rules menu. The timeout is the time the players have to finish the scenario. Basically, if a player is not able to finish the scenario in the specified

time to complete, he may loose points and move to the next scenario, be redirected to a different scenario than the players who finish it on time, or even immediately loose the game, depending on the rules and flow defined by the developer.

With regards to the flow related to the current scenario, it can be customized in the Flow menu of the Scenario properties tool-bar. Here the current scenario can be set as the Initial Scenario of the game and all the transitions and flow related to the current scenario can be customized, regarding when the timeout defined is over, the scenario is skipped or finished. Flow by Knowledge can be enabled, referring to the percentage of total score obtained at the activities of the current scenario by the players, according to their performance. When using two knowledge divisions, the flow differs if the score is between 0%-50% or 50%-100%. Three knowledge divisions indicates a flow divergence between 0%-50%, 50%-90% or 90%-100% of the total score.

- **Element properties**: When an element is selected in the scenario preview area or through the Explorer side bar, whether it is a resource, shape or activity, the properties tool-bar for the current element will be displayed up. Regarding the element is not an activity, text content inside the element and background color change is allowed. However, these two properties only apply to Shapes, Headings or Paragraphs, once there is no point in changing these properties when the element is an image, PDF, video or audio. Moreover, the Text menu provides a way to customize the text content inserted, allowing to change its size, weight, alignment and color. The Dimensions menu lets the developer manually change the width and height of the current element. A more common way to change the element's dimension is to click on its edges at the scenario preview and resize it. The current X and Y values in pixels of the position of the select element can be seen at the toolbar, and may be changed by selecting the element and dragging it inside the scenario preview area. Additionally, element's border thickness and color may be changed. Similar to the text content and background color, border properties changes only apply to Shapes, Headings and Paragraphs.

- **Activity properties**: On the other hand, if the element is an activity, the activity properties tool-bar will be the one to be displayed (Figure 3.32). Each activity has available a set of sub-activities according to its type, which can be edited and included in the game together with other activities from different types. For each of the activities included in the game it is possible to define helps, that can only be used in game taking into account the number of helps already used in the scenario. Moreover, each activity has a final score, inherit to that activity, which can vary according to the rules set by the developer and the performance of the player. The more helps used and wrong answers given by the player, the less the score is. Despite the developer rules for the activity, all the activities have a similar weight

in the final score of the game. All things considered, there is the Open Activity builder feature, which provides a way to edit the current activity by redirecting to its specific Development Environment. To notice that for each type and sub-type of activity there is an appropriate builder, but for all of them there is a button to redirect the developer back to the project he was building before going to the activity Development Environment.

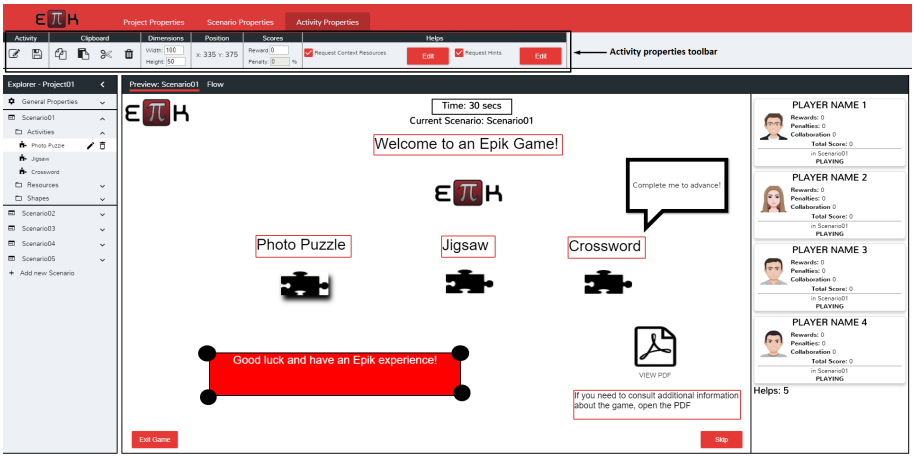

Figure 3.32: Activity Properties Toolbar at the Project Builder.

Similarly to the elements, activities also have a Clipboard, Dimension and Position menu at its properties tool-bar, which provide the same functionalities as the ones from the elements tool-bar. Moreover, here the rewards for completing an activity with a flawless score of 100%, meaning that no penalties were added, can be defined, as well as the penalty % the reward value will suffer if the players can't complete the activity within the activity rules defined. Lastly, the helps available for the current activity may be included according to two types: Context Resources and Hints. Context Resources are resources which will provide information related to the activity to help the players understand its context. Any resource available at the current workspace can be used as a Context Resource. Hints may be only a text paragraph giving a tip to the player, or a resource, similar to the context resource. However, hints provide an help to complete the current activity and not just information about its context. Both of these types of helps count as an help used in the scenario and may bring penalties to the players, depending on the game rules defined by the developer (Figure 3.33).

All things considered, through the Project Builder the developer has the ability to create, develop and customize all the aspects he wants to have in his game. Yet, before being ready to be imported to the Epik Website, the final project needs to face validation and exportation.

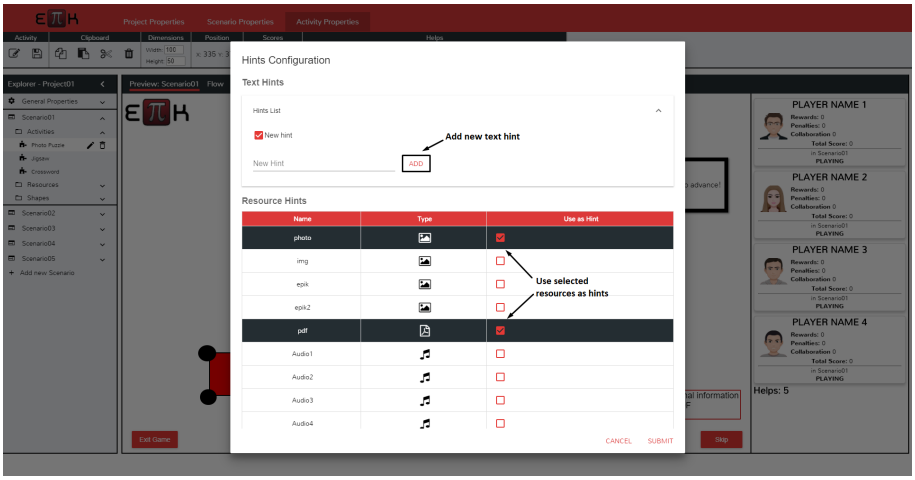

Figure 3.33: Hints Menu at the Activity Properties Toolbar. Context Resources menu follows a similar structure without the Text hints component.

3.2.3 Epik Games Validation and Exportation

With all being said, when all the content of the project is edited, it will need to be exported as a game in order to be eligible for import at the Epik Website and become a playable game. However, before exporting the project, some verifications have to be made in order to check if the project is valid. Taking into account the previous version of the Epik platform, game validation and exportation conditions didn't suffer major changes. However, there is a case where the Scores and Rankings Scenario might not be seen by the players, which happens if they exit the game or disconnect due to going back, closing or refreshing the browser when the game is executing. If this situation occurs, the only one that can see these players scores and ranking is the developer, when checking the logs related to the respective game session. Moreover, on this new version of the platform, games might not contain activities, as previously referred, so the conditions for game validation regarding activities inclusion and associated helps are no longer effective. Despite this fact, in case the game has activities included in its scenarios, there is a need to verify if all of them:

- Have a sub type associated, meaning that after being created a sub type must be chosen;
- Are validated at their own Development Environment according to the sub type associated.

Overall, if the project gathers all of the referred conditions, it is ready to be exported as a playable game. The option to export the game can be found in the project properties toolbar at the Project Builder. While it is not valid, if the save or export option is clicked, the developer will receive a notification that the game is not valid. In case the save option is selected, changes will be saved despite the game being valid or not, and a notification is

send regarding which of the conditions referred above is not fulfilled (Figure 3.35). The export feature is also available at the projects area of the Dashboard. In either situations, the application will verify if the project gathers all the conditions to be exported, and when it is validated, the developer is notified that the project was exported successfully (Figure 3.34. This action will create a .ZIP folder, including all the resources files related to the game content, and a .JSON file, containing all the information of the generated game, regarding game rules and scenarios content and flow. The .ZIP folder is stored at a selected path of the developer's operative system. Additionally, a row containing information about the newly generated game will be created in the table at the Games area of the Dashboard. The name given to the generated game will be the same name the developer previously gave to the exported project. Yet, this name can be changed by accessing the edit option of the Games table. The generated folder can be later on imported to the Epik Website by the developer in order to make it available to be played, a subject which will be discussed further on in this chapter.

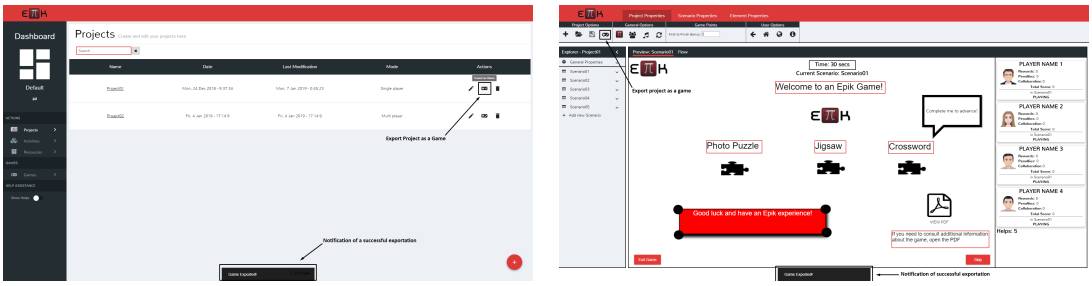

Figure 3.34: Successfully Export a Project as a Game at the Dashboard (on the left) and at the Development Environment (on the right)

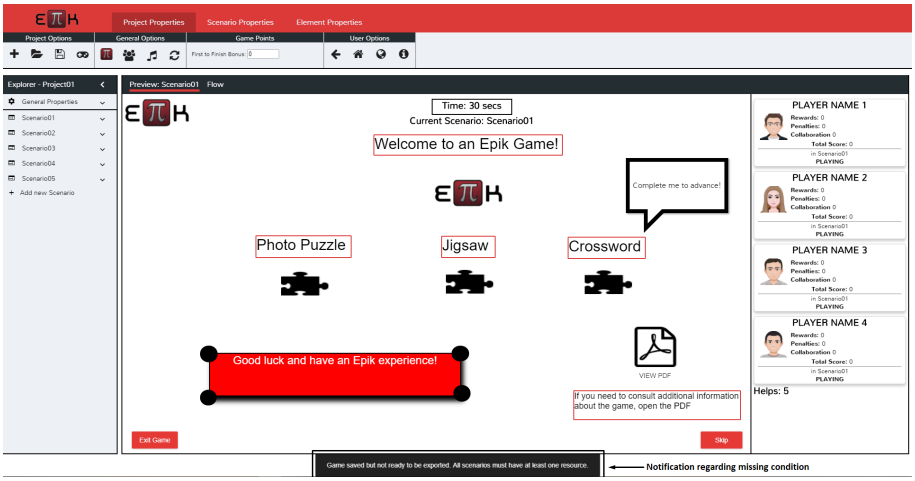

Figure 3.35: Missing Condition for Validation after saving Project.

3.3 Interactive Puzzle Activities Development

The activities are the interactive part of the Epik Games. Assuming that the developer has downloaded and installed the Puzzles plug-in in the Desktop Application, the creation of new puzzle activities becomes available. Once the activity is created in a certain workspace, either at the Dashboard in the Activities section or at the Scenario properties in the Project Builder, its Development Environment, the Puzzle Builder, will be available to allow its edition. This environment can be entered when selecting the activity at the referred section in the Dashboard or on the 'open activity builder' option at the activity properties in the Project Builder, which appears when that specific activity is selected in the current scenario (Figure 3.36).

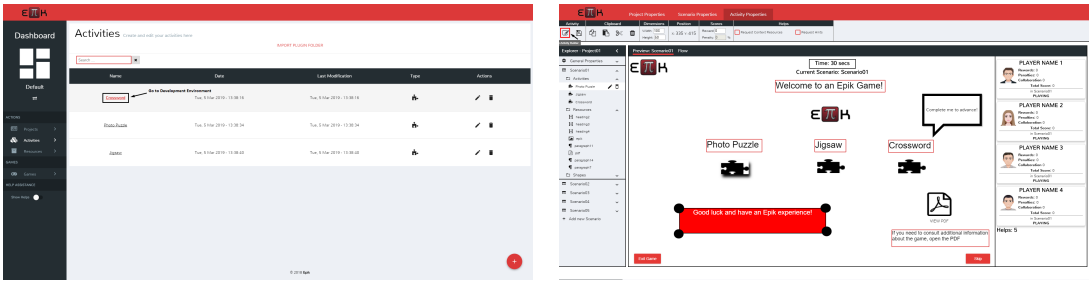

Figure 3.36: Opening Puzzle Builder at Dashboard (on the left) and on the Project Builder (on the right)

When at the Puzzle Builder of the Interactive Puzzle activity (Figure 3.37), the developer may choose between three sub types, that is, three different puzzle games, from Crossword, Jigsaw or Photo Puzzle. If, for some reason, the developer feels the need to make this choice later, he may go back to the last page he was on, whether he entered the environment from the Project Builder or from the Dashboard, or go directly back to the Dashboard. However, without a sub type, even if this activity is included in a scenario of a project later on, the project will not be valid and, therefore, available to be exported as a game.

Besides, upon the sub type of Puzzle selection, depending on the chosen sub type, a different menu will appear for configuring some aspects of the puzzle game before starting its edition. After these configurations are done, the developer is redirected to a specific Development Environment, according to the selected sub type, in order to edit the activity. The available Development Environments and its features will be described in the following sub-sections, according to each sub type. To notice that the next time the developer goes to the edition of an activity, if it has already a sub type associated, it will redirect to the specific Development Environment instead of to the Puzzle Builder, as a sub type was already previously defined for that puzzle activity.

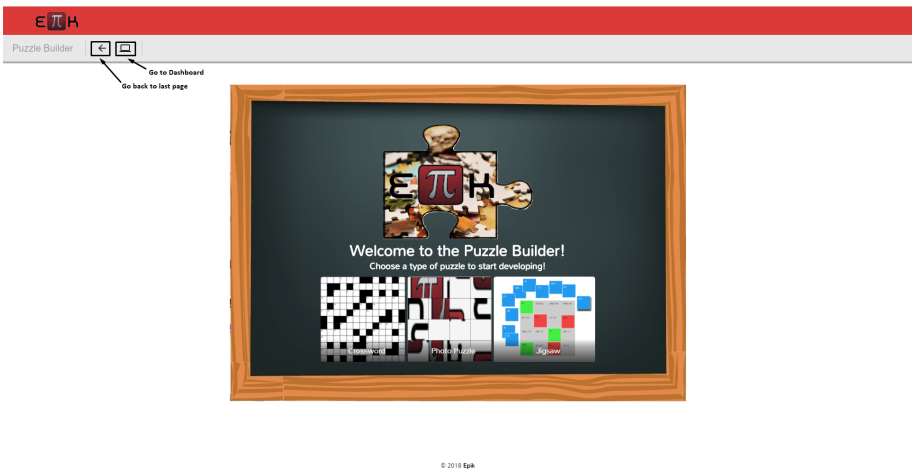

Figure 3.37: Puzzle Builder of an Interactive Puzzle Activity.

3.3.1 Crossword Puzzle

The Crossword Puzzle is an interactive representation of the regular crossword game, where a list of clues across and down are provided to the players and their goal is to fill all the empty squares of the board with the correct solutions to the descriptions of the clues. This was one of the select types of puzzles to develop was the wide diversity of educational subjects that could be included in the game context and the fact that it helps reinforce terminology and improve players general knowledge.

With that in mind, when the selected sub type refers to a Crossword Puzzle game, at the Puzzle Builder, a menu will appear to allow the developer to choose the size of the crossword board, from a 5x5 to 13x13 matrix. After selecting the board size, the developer will be redirected to the Crossword puzzle Builder. At this Development Environment, a preview of the current puzzle being edited is presented, containing an empty board with the previously selected size, an empty list of the across and down clues and a toolbar with edition options.

Regarding the crossword board, it will contain all the solutions for the clues across and down on the list, placed in the selected position and filling the board according to the solution size. Each square containing a starting letter of a solution will have a number to indicate to which clue the solution belongs to. Additionally, the letter present in each square may belong to a clue across and/or down, but may only be part of a maximum of two solutions in the board. Also it is only possible to add a clue across and down with the same number if they start in the same square with the same letter. Other verifications are made when building the crossword in order to avoid the developer to commit unnecessary mistakes. The list of clues is divided in two columns, one for the clues oriented across and other for clues oriented down. Each clue on the list will have a number indicating the correspondent solution in the board. Additionally, when editing, the clue in the list will be highlighted if the squares of the board containing parts of the correspondent solution are being hovered by the developers cursor. This feature will help to have a

better visualization of what is being edited, specially when some clues and solutions are already insert at the crossword board.

At Execution Environment, the solutions for the clues will be omitted and only the squares containing a starting letter of a solution will have the corresponding clue number displayed. When players insert a letter in a certain square, if it corresponds to the letter of the solution, the square will turn green, indicating that the letter is in the correct position. On the other hand, if it doesn't correspond, the square will turn red to indicate failure. When all the white squares of the board are green, the activity will be complete and the rewards according to the score % obtained will be added to the player score.

Furthermore, the toolbar will allow to build the crossword board content and respective clues, as well as save the progress and navigate to other pages. The first section of the toolbar provides a way to go back to the previous page the developer was in or go directly to the Dashboard, similar to navigation options at the Puzzle Builder. Additionally, in this section it is possible to restart the crossword, which will clear all the board content as well as the clues list, and save all the changes made. When saving the game, a notification will be shown, indicating that the changes made were saved and if the activity is valid. In order to have a valid Crossword puzzle, all the squares of the board must be filled whether with a letter or a black square, otherwise the activity won't be valid (Figure 3.38). Despite not being valid, the changes made are still saved, but the activity will make the exportation fail if included in future games.

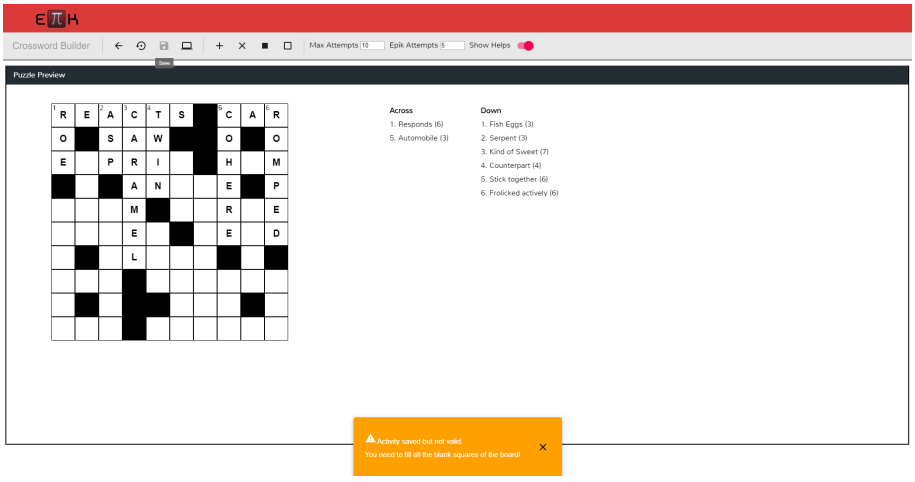

Figure 3.38: Invalid Crossword Activity at Crossword Builder.

With regards to the second section, it will allow the building of the crossword. The plus button lets the developer add a new clue and respective solution in a certain position with a number associated (Figure 3.39). If there are no clues and solutions added to the crossword, the remove and restart icon will be disabled. Otherwise, the first referred icon will allow the removal of the selected clue. Furthermore, it is possible to add a black square and remove it by selecting the square of the board at the last two options of the section. This same functionality can be achieved by clicking in the desired square with

the right click of the mouse. A right click on an empty square will turn it into a black square and the same action on a black square will make it empty again and available to contain a letter of a solution.

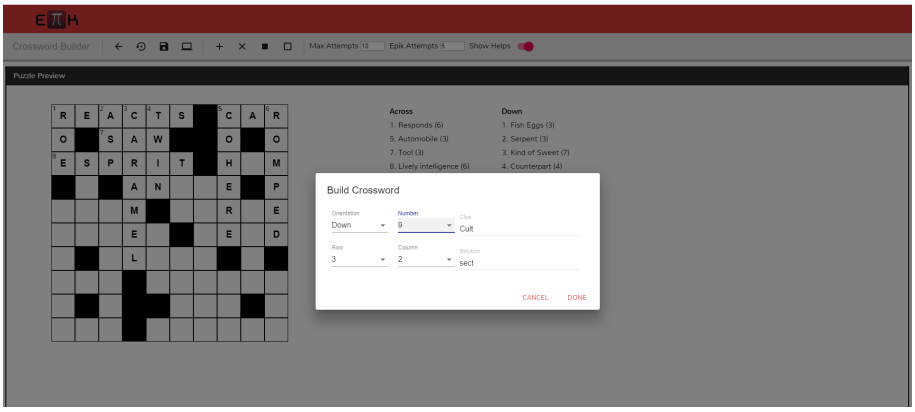

Figure 3.39: Crossword Builder menu for building Crossword.

The last section of the toolbar is related to the score and attempts the player will have during the game. Here the developer must define the number of attempts, called Epik attempts, the players will have to complete the puzzle and get a score of 100% of the points associated to that activity in a certain scenario. An attempt in the Crossword Puzzle's context is intended to be each time the player inserts a letter in a square and that it is not the correct one for that square, that is, each time a square turns red in game execution. For each additional attempt, a score % will be lost. Moreover, the number of Max attempts must be defined, which is the number of attempts that when reached, the game will be over and the final score % of the activity will be 0. The tracking of these score % will be displayed during the game execution. To finalize, it is also possible in this section to enable/disabled the Epik manual by toggling the switch. When all editing is over and the activity is valid, it is possible to export a game including this activity (Figure 3.40).

3.3.2 Photo Puzzle

The Photo Puzzle, as the name suggests, consists on an image divided into pieces, which are positioned at random positions. The objective of the game is to place the pieces in the correct positions, by having available the final result image or hints which will help the player to visualize what the final result might be. This puzzle was selected to be developed as it gives freedom to the developer to include any image he wants as a puzzle, regarding any educational theme he wants to approach. Furthermore, it helps improve players reasoning and visual processing skills. With that said, the selection of a Photo Puzzle game will trigger a menu where the puzzle level must be chosen, from level 3 to 6, which represents the number of divisions the selected image will have. For example,

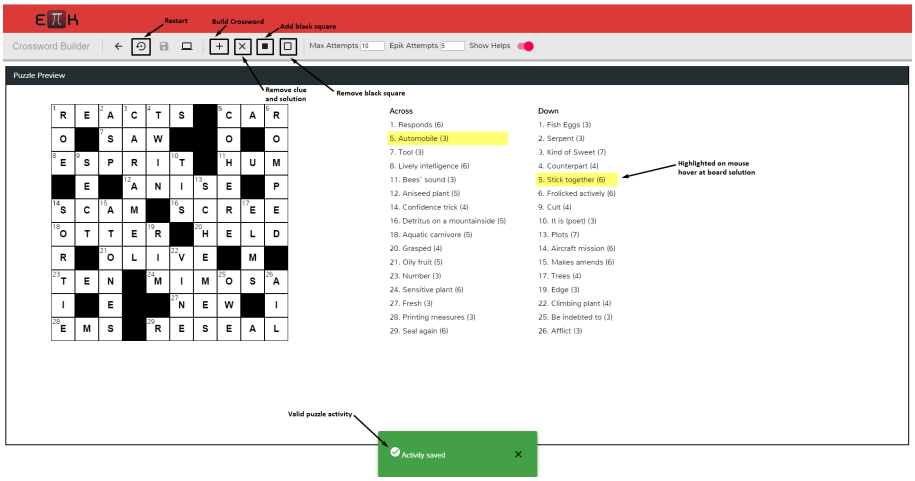

Figure 3.40: Crossword Builder Interface.

if the puzzle level is 5, the image will be divided in a 5x5 grid, meaning that the puzzle will have 25 pieces.

After selecting the puzzle level, the Photo Puzzle Builder will be displayed, where a preview of the current puzzle being edited is presented, containing a mixed default image given just for testing, which is why some of the toolbar functionalities are disabled initially, until an image is assigned to be the puzzle photo. On the left there is available a side bar with the current puzzle photo complete to help guiding in the construction of the image, and the option to change the puzzle photo (Figure 3.41). When changing the puzzle photo, a list of the available images from the current workspace is displayed, allowing to select one of them as the puzzle photo or to add a new image to the current workspace that may be selected as the puzzle photo as well after its creation. This functionality may also be found at the Photo Puzzle builder toolbar. Having this feature available in two different places may be useful, specially in case the developer hides the side bar for a better visualization of the puzzle preview (Figure 3.42).

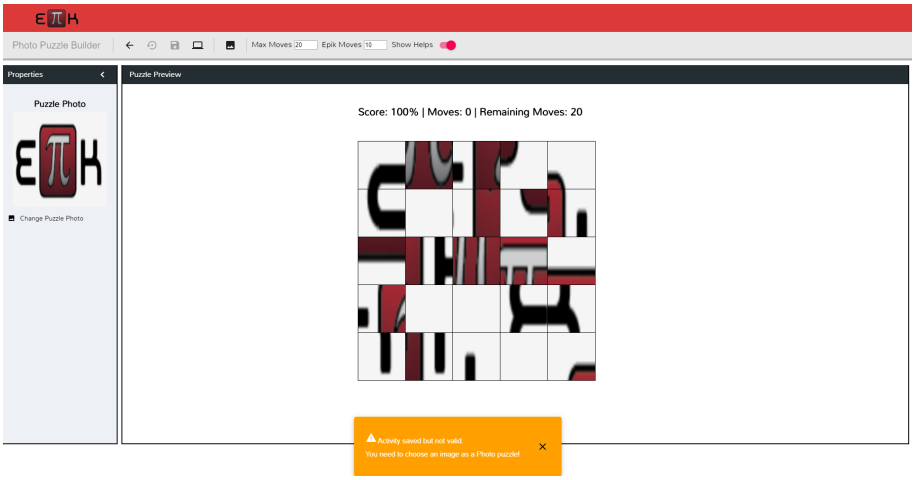

Figure 3.41: Invalid Photo Puzzle Activity at Photo Puzzle Builder.

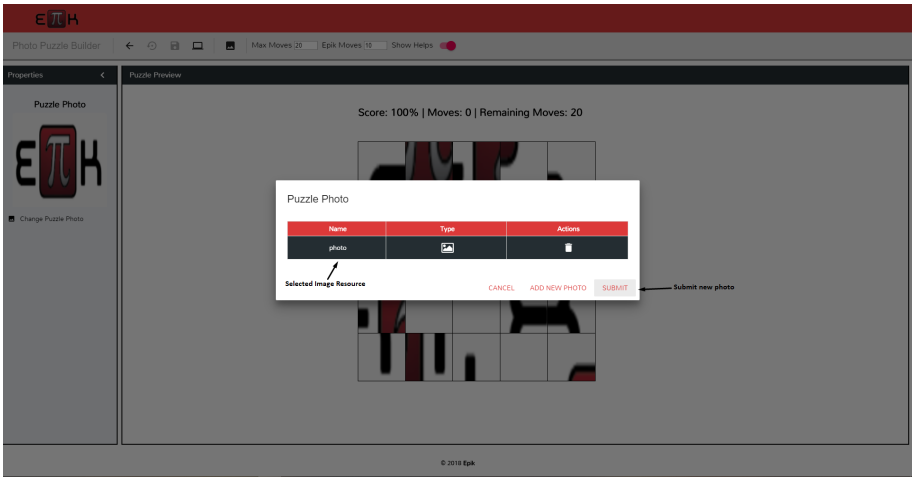

Figure 3.42: Photo Puzzle Builder menu to change Puzzle Photo.

Above the puzzle area it is possible to see how many moves the player has used so far and the remaining moves he still has according to the maximum number of moves defined to complete the puzzle. A move is intended to be each time the player changes the position of a piece in the board. In case the player finishes the puzzle within the range of Epik moves defined, his score will be 100% of the points associated to that activity in a certain scenario. For each additional move the score % will decrease and in case it reaches the Maximum moves defined, its score % will be 0 and the puzzle game ends. At Execution Environment, the epik moves limit for completing the activity will be displayed, as well as the total reward attributed to the activity.

Lastly, the toolbar's first section offers the same functionalities as the ones from the previous Puzzle game described, apart from the effects of the restart feature, which in this case will reset the puzzle photo and all the moves made and score, displayed for testing. Moreover, when saving the changes made, the validation of the puzzle activity is also verified. In case the puzzle activity is a Photo Puzzle, the only requirement for being valid is to have an image associated other than the default image given initially for testing purposes. Finally, the display of helps from the Epik Manual may be switched on/off aswell. Having the puzzle photo changed and configured all the moves limit, the activity may now be included in a valid game (Figure 3.43).

3.3.3 Jigsaw Puzzle

The Jigsaw Puzzle is composed by a board and a set of pieces. The board may contain the answers and the pieces the enigma, or vice-versa. The objective is to fit the pieces in the correct places of the board. This was selected since is one of the most common types of puzzle and adapts to almost any type of subject, developing different types of cognitive skills of the players regarding the game created. Moreover, when this Puzzle sub type is selected, a menu will be displayed to define the size of the board, from a 3x3 to 6x6 grid.

Defined the size, the Jigsaw Puzzle Builder will appear, showing a preview of the

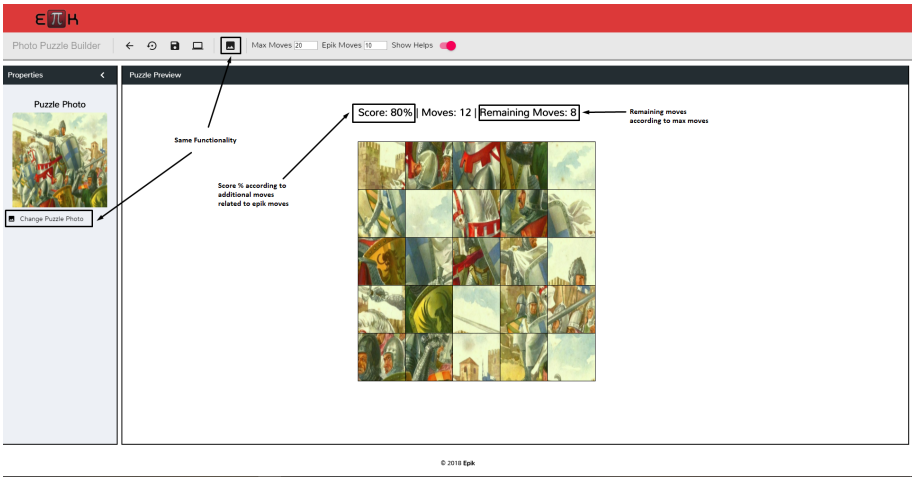

Figure 3.43: Photo Puzzle Builder Interface.

current puzzle being edited, which initially will have all the board fittings with the same name but distinguished by a number, and all the pieces with the same nomenclature. The pieces will be randomly displayed in the page, but each piece will match with the board fitting with the same number present in the piece name. When a piece is outside the board, its color will be neutral, meaning that it hasn't been associated to any fitting of the board. If the piece is dropped in the correct position, its color will turn green. On the other hand, if dropped in a wrong position, its color will turn red (Figure 3.44).

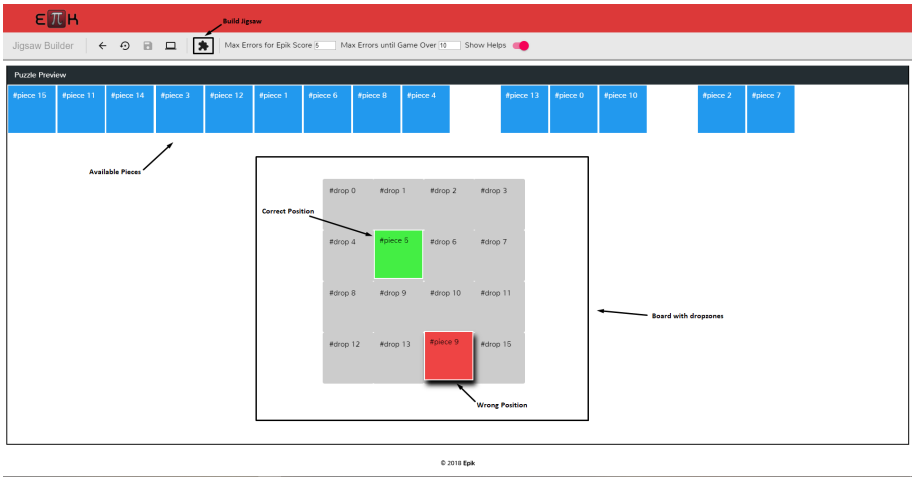

Figure 3.44: Jigsaw Puzzle Builder Interface.

In order to build the puzzle by editing the content of the pieces and correspondent fittings in the board, there is an option at the toolbar which will open an edition menu. After choosing the number of the piece and dropzone to edit, it is required to define the content of the piece and correspondent fitting, where there is the possibility of one containing the enigma and the other the answer or vice-versa, as told before. This flexibility will allow developers to edit the puzzle according to a wide variety of educational themes, such as mathematical expressions, where there are expressions and answers in the pieces

and dropzones of the board, or historical figures, with a short description and historical figure name distributed along pieces and fittings, for example (Figure 3.45).

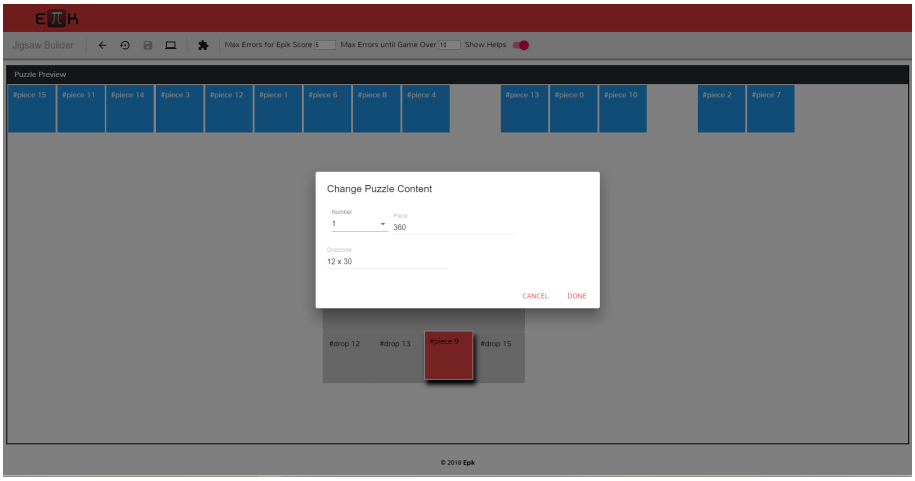

Figure 3.45: Jigsaw Puzzle Builder menu for building Jigsaw.

Similar to other puzzle activities, the toolbar's first section offers the same functionalities as the ones from the previous Puzzle games described, apart from the effects of the restart feature, which in this case will reset the content of the pieces and board. Also the validation is checked after saving the changes made, which in this case requires that all the pieces and board dropzones have its content edited, that is, different from the default one given. The helps from the Epik Manual can be toggled on/off as well. After finishing editing all pieces and dropzones content, the puzzle activity will be valid and ready to be included in a game (Figure 3.46).

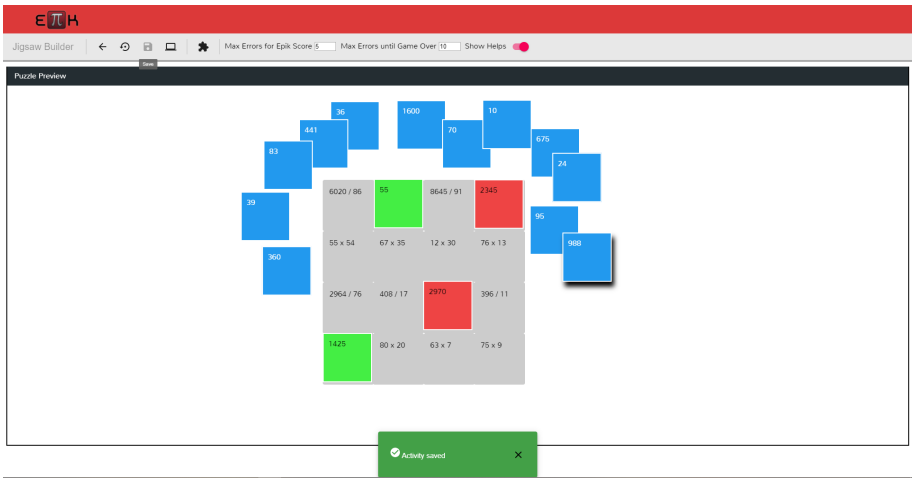

Figure 3.46: Valid Jigsaw Puzzle.

3.4 Execution Environment

Apart from the Desktop Application, responsible for the development of the Epik Games, the Execution Environment may have two different participants: developers and players. In a brief description, developers will be responsible for posting the Epik Games, making them playable for certain players they associate a game to. This environment is available through browser, and includes an Website and a games execution area, which will be described in detail on sections 3.4.1 and 3.4.2, respectively.

3.4.1 Epik Website

Taking into account the Epik Website from the previous version of the platform, the new Website offers more features, regarding the addition of new components. Here the users will find a brief description of the Epik platform in general, as well as its main features, specifying the new activities. Having in mind the requirements stated at the beginning of the chapter, there is also a Download area from where the developer can obtain the Epik Desktop Application and the plug-ins with the different type of activities to be included in the games developed in the Desktop Application.

In order to import a playable game or to just start playing an Epik Game, there is the 'Enter Epik' option (Figure 3.47). When in there, according to the pretended a action, users must choose between Developer and Player area. As a developer, as told briefly, the user is allowed to upload games he previously created at the Epik development application to be played by certain players of his choice, whereas as a Player the user may only play games that he was assigned for (Figure 3.48).

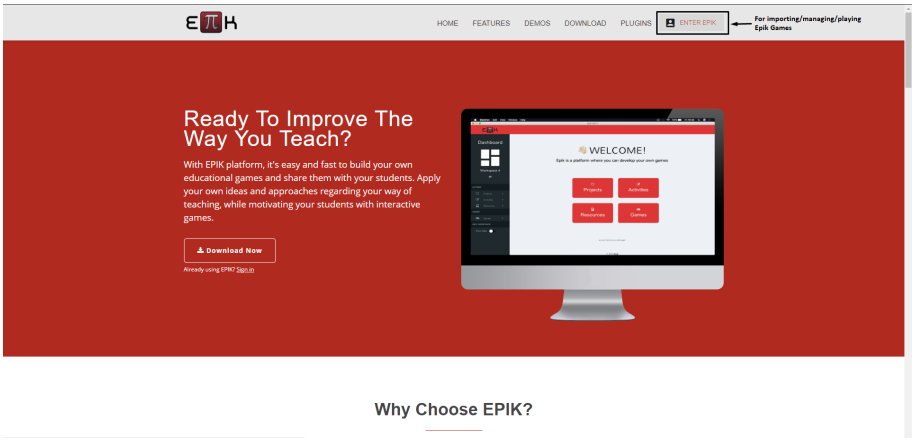

Figure 3.47: Epik Website option to go to Execution Area.

Regarding the user is a developer, before uploading games an account creation and login to the Website is required, in order to save information related to the developer and the games he uploads. In terms of the Register page (Figure 3.49 on the left), comparing to the one from the previous version of the platform, the terms and services was too extensive. With that in mind, the solution thought was to provide a way to toggle on

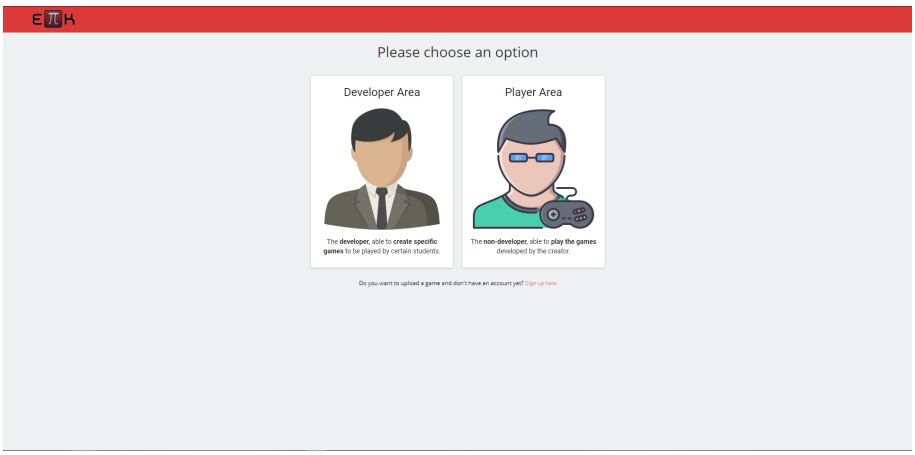

Figure 3.48: Epik Website Page to choose between Developer and Player Area.

and off the terms and services in order to make the page look cleaner (Figure 3.49 on the right). After registering, users may log in as developers. The Login page (Figure 3.50 on the left) structure had no problems, meaning that in the new version of the platform the page only had a design change (Figure 3.50 on the right). When logging in, a dashboard will be displayed, where new games can be imported, profile information can be edited and the list of games uploaded by the developer to the Website is shown, as well as games that other developers have assigned him to play. Therefore, a developer may also be a player, since he can play is own games and games he is assigned to by other developers. However, despite being a developer, he is a regular player in regards to that specific game he hasn't developed but was assigned to play, meaning that he can't edit or consult session logs related to that game.

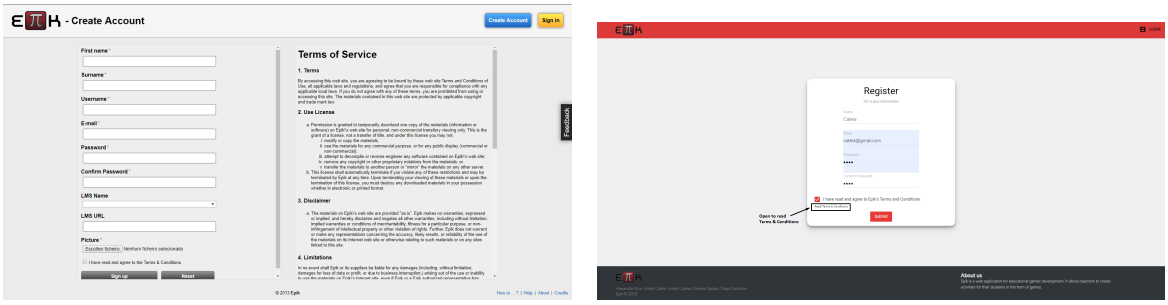

Figure 3.49: Epik's previous version (on the left) and new version of the Register Page (on the right).

When uploading a game, the imported folder should be a .ZIP folder exported from the Epik application, which should contain a valid game and associated resources. Additionally, a start and end date should be defined, meaning that the uploaded game can only be played between those dates, as well as a game password. To note that these players, as referred previously, may also be other developers. All the referred game configurations can be edited after uploading the game. Having all set, the game may be uploaded to the

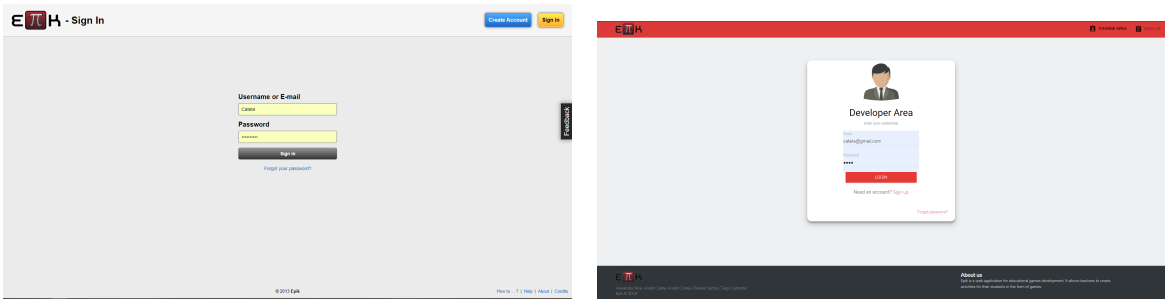

Figure 3.50: Epik's previous version (on the left) and new version of the Login Page (on the right).

Website. After the game upload, a new row with the game details will be added to the table at the Dashboard. At each row, it is possible to see the name, mode, creator, start and end date of the game, and get access to actions related to it, if it is a game uploaded by the developer. These actions include deleting the game from the Website, playing the game and consulting session logs. The session logs regarding Single Player games contain information about the players performance regarding all sessions of that game, as well as the number of sessions played.

From another perspective, as a Player, there is no need to create an account. Instead, an email address must be provided to the Website to be used as an identifier. Upon entering the email, similar to developers, players will be redirected to a dashboard, but will only have a list of games that they were assigned for through email by developers. Also apart from developers, players are not allowed to upload any game to the Website, as well as editing their profiles, since there is no account associated. To sum up, players role in the platform is to play the games they are assigned for, meaning there is no need to keep any additional information apart from their email, final scores and performances in games.

With this in mind, in order to start playing, players must select a game from their list. Before running the game, the Website requires the game password to make sure that the user who is playing the game is the one whose email was in the game's list of players. The same process will occur in case a developer is assigned to a game from other developer. In this particular cases, developers behave as regular players. Having introduced the correct game password, the game will start.

3.4.2 Epik Games Execution

When inside an Epik Game, the first scenario displayed is the Start Scenario, already referred previously. This scenario is common to all games, where players may exit the game, check the instructions to know more about the game before starting, and start the game. The instructions scenario will give a brief notion of the general game rules which apply to all Epik Games. Before the game starts, players are redirected to an waiting room,

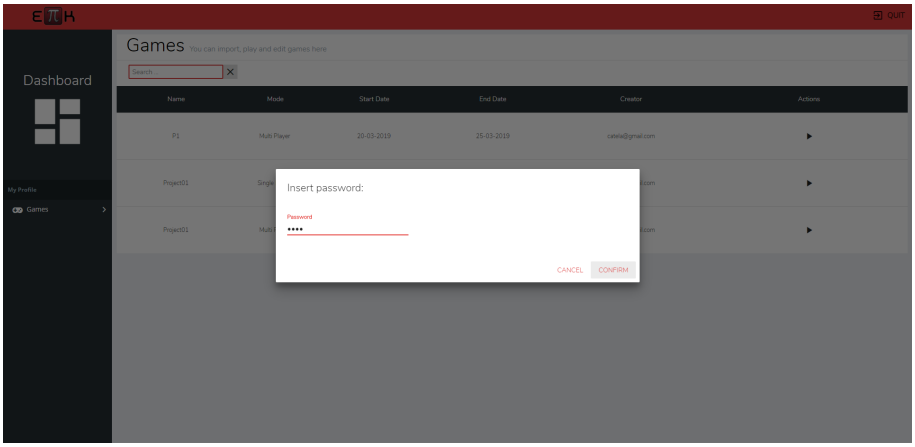

Figure 3.51: Player inserting Password in order to Enter the Game.

where they must choose their name and avatar in game, which will appear at the player section in each scenario 3.52. In Single Player mode, the game will start after clicking on the play button. However, if the game mode is Multiplayer, each player must wait for the remaining players to connect. In case the time of waiting for other players to join expires, if the minimum number of players defined by the developer is achieved, then the game starts even if not all the players have joined. However, if the minimum number of players is not reached, when the time is over the game will not start and all players must join again and wait for at least the minimum number of players to be reached in order to start the game 3.53.

Figure 3.52: Selecting Name and Avatar for the current Game Session.

Regarding the gameplay itself, whether the game mode is Single or Multiplayer, it is possible to have a game without any activities, that is, without any practical strand. In fact, the Epik Games objective is also to offer developers a way to teach and give more interactive lessons to the players, in order to capture their attention more easily. Yet, the majority of Epik Games will have activities in their scenarios. According to the activity type, a different icon will appear and by selecting it, a window will open, containing

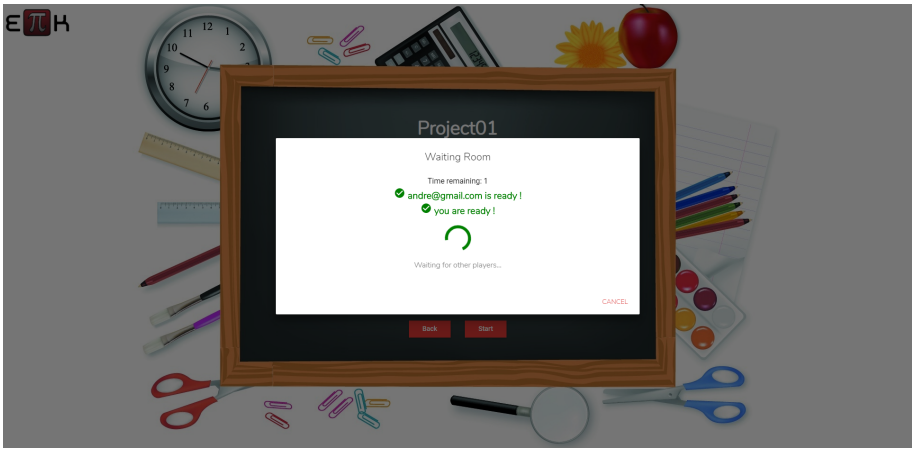

Figure 3.53: Waiting for Other Players to connect the Game.

the activity to complete. When the activity is an Interactive Puzzle activity, as shown in Figure 3.54, the icon displayed will be a puzzle piece, and the window may contain any of the three types of puzzle described in the section 3.3. While playing a puzzle activity, for example a Photo puzzle (Figure 3.55), players may complete it with 100% of the score and get the total reward defined for the activity, or lose score % according to his performance. In case the player wants to leave the activity, by selecting the 'finish activity' option, his reward will be zero. In either situation, the puzzle activity will end and the respective window will close. The current scenario the player is in is displayed again, and the score obtained in the activity, if the player manages to complete it, will be added to the player card 3.56.

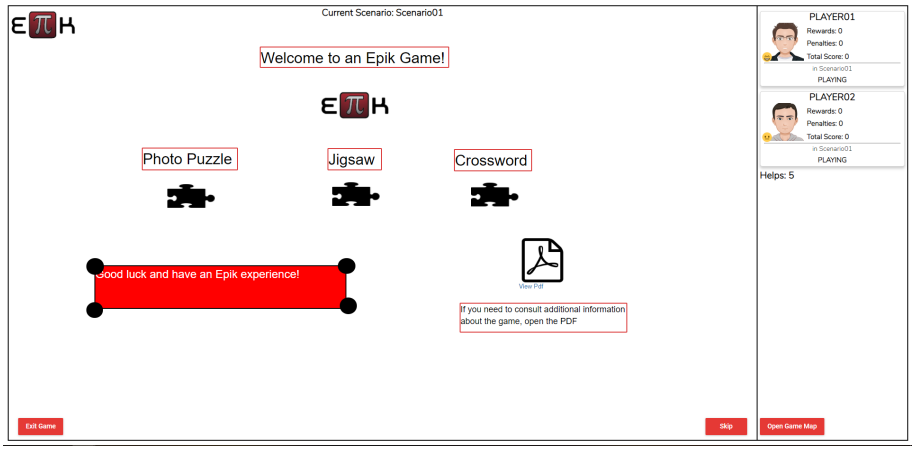

Figure 3.54: Example of a Game Execution.

With regards to the other two Interactive Puzzle activities developed, the Jigsaw Puzzle was already playable at the Desktop Application for testing purposes and the display at the game execution is very similar (Figure 3.45), only having the information about the game status as seen in the previous example from Figure 3.55. The Crossword Puzzle, however, was not playable at the Desktop Application since the developers needed

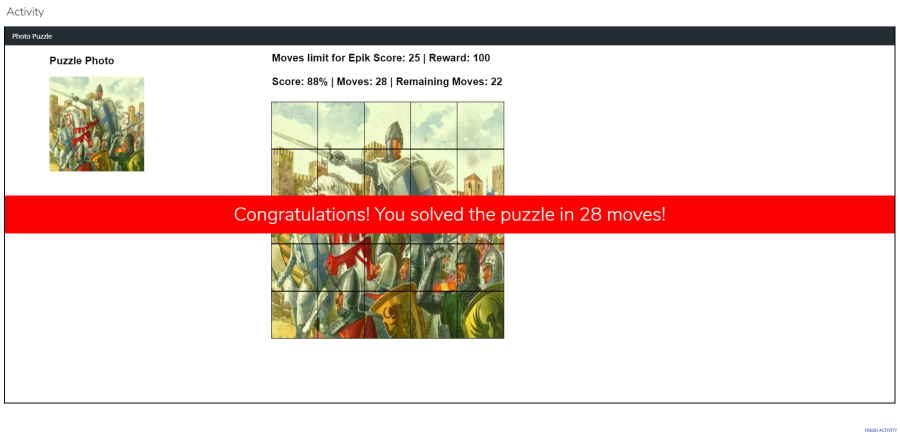

Figure 3.55: Completion of a Puzzle Activity.

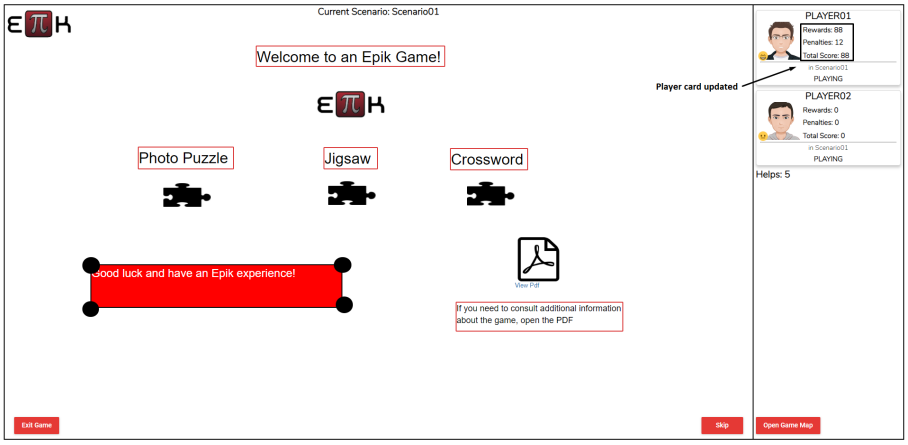

Figure 3.56: Players Card updated after finishing Activity.

to be able to see what content they were adding to the board. Thus, for the execution environment, players must fill the empty board with the solutions regarding the clues given, as shown in Figure 3.57.

Returning to the game play, in order to finish the game, players must complete all the scenarios from a flow or flows that lead to the end of the game, within the game rules defined by the developer. During the game, there is always available the game map, where the flow between scenarios is displayed. Each scenario may have a restricted time to be completed and a limited number of helps that can be used by the players in the activities of the scenario. If any of these rules is not followed, players may face penalties. Depending on the game rules, by not ending the scenario in time for example may lead to penalties in the final score but still allow players to keep playing and follow the same flow or be redirected to other scenario flow, whereas sometimes not finishing the scenario may end in a game over.

Additionally, in Multiplayer mode, the players state in the game is always displayed. This state can be, besides playing, lost the game, disconnected, give up or finished the

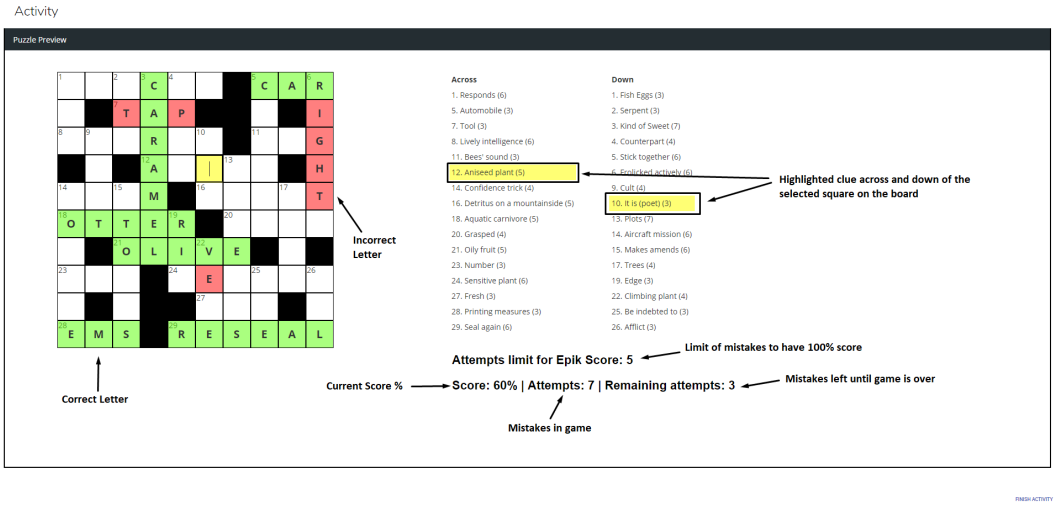

Figure 3.57: Crossword Puzzle Activity Execution.

game. Moreover, players can be in different scenarios at a given time, a feature that is new regarding the previous version of the platform. In the previous version, players were always on the same scenario at a given time and had to wait for others to complete the scenario in order to move to the next one. Having this in mind, at the Player Section is possible for each player to see other players state and the current scenario they are in, as well as their scores. When finishing a game, in a Single Player game, the player is redirected to the Score scenario, where his final game statistics are displayed 3.58. In a Multiplayer game, players who finish a game are redirected to the Finishers room, where their final game statistics are displayed, but they have to wait for other players to finish in order to see the final rankings 3.59. When all players have finish the game or even if some have disconnected or lost the game, the Rankings scenario will appear.

Figure 3.58: Scores Scenario for Single Player.

Players who lost the game will still have their score displayed until the moment they lost, but they will always be ranked before the players who finished the game, despite

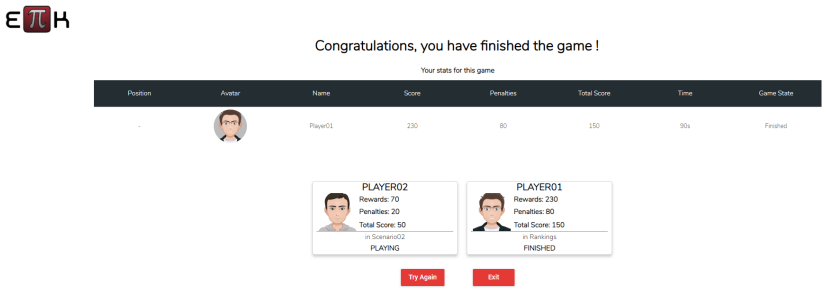

Figure 3.59: Finishers Room in a Multiplayer game.

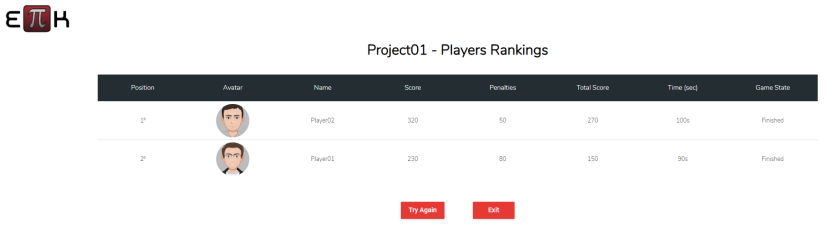

Figure 3.60: Rankings Scenario at a Multiplayer game.

the score. Moreover, players who got disconnected or exited the game won't have any score displayed in the scores and ranking scenarios. Each time a game session is played, the information regarding players involved in the game session will be available at the correspondent developer's dashboard 3.61.

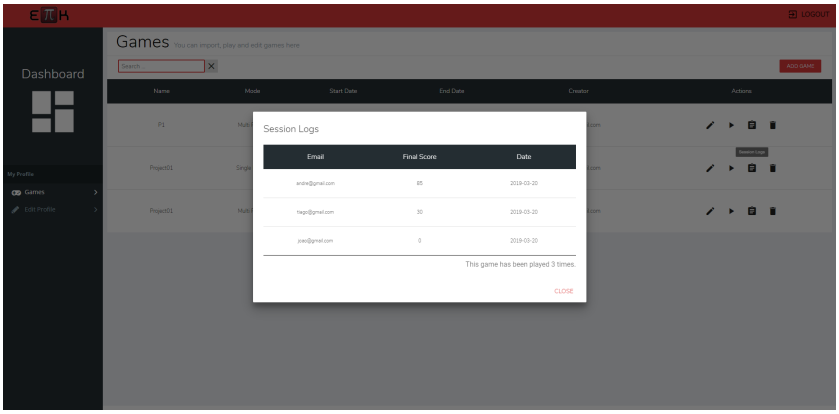

Figure 3.61: Logs regarding Single Player Game Sessions.

3.5 Chapter Summary

Throughout this chapter is presented all the proposed solutions regarding all the Epik platform common work and Interactive Puzzle activities development and execution. As a sum up, a scheme regarding all the process described is displayed in Figure 3.1. The content of this chapter will serve as a base to what requirements the frameworks used to implement the platform and Interactive Puzzle activities needed to have and also implementations decisions regarding what is pretended and the frameworks limitations, which will be discussed in the next chapter.

Chapter 4

Implementation Options and Decisions

The presented chapter includes how the Desktop Application and Execution Server, as well as the Interactive Puzzle Activities, are implemented, having in mind the options and decisions taken related to the Database model, Architecture style and Frameworks for building them. Based on this last topic, the choices of framework to meet the requirements of the proposed solutions for the implementation of the new version of the platform described in the last chapter are enumerated.

4.1 Frameworks Selection

Taking into account the main changes discussed in the previous sections, to develop the Desktop Application that will be used for creating and managing the Epik Games, including its activities and the other components associated to it, a framework was needed that could provide:

- **Known Programming Languages** : By having programming languages in which all students involved in the project were familiar with, the development becomes easier;
- **Ease of use**: To facilitate the work from the students involved in the project in building the Epik platform;
- **Portability**: To be able to support different Operating Systems in order to make it accessible for a bigger amount of users;
- **Database Support**: In order to be able to store the content regarding game content, activities, resources, activities, among others;

Table 4.1: Display of Programming Languages and Supported Platforms of each of the searched Frameworks for the Development of the Desktop Application

Framework	Programming Languages	Supported Platforms
React Native	JS, React	Windows/MacOS/ /Android/iOS
Electron	JS, HTML and CSS	Windows/Mac OS/Linux
Haxe	Android, PHP, Python, C#, Java, NodeJS	Windows/MacOS/ /Android/iOS
Node Webkit	NodeJS	Windows/Mac OS/Linux
Xojo	Xojo (similar to VB, Java and C#)	Windows/MacOS/Linux
Enyo	JS, HTML	Windows/MacOS/ /Android/iOS
8th	8th	Windows/Mac OS/Linux/ /Android/Rasp/iOS

Table 4.2: Display of Other Characteristics of each of the searched Frameworks for the Development of the Desktop Application

Framework	Ease of Use	Database Support	Paid
React Native	3	Yes	No
Electron	2	Yes	No
Haxe	1	Yes	No
Node Webkit	4	Yes	No
Xojo	6	Yes but difficult	Yes
Enyo	7	Yes	No
8th	5	Yes	Yes

- **Free use**: If it's possible for the framework used by the students for the development of the Epik platform to be free of any cost, producing similar results, paying unnecessarily should be avoided;

Having this in mind, a search was made for available frameworks, and a comparison was made in order to choose one that gathers the best conditions regarding the operations we want to perform. That said, follows two tables with the comparison between the searched possible frameworks for the Desktop Application development. The table was divided in two in order to be more understandable, since gathering all the characteristics searched in one table would not fit into the format of this thesis.

Regarding both Tables 4.1 and 4.2, most of the frameworks searched gather a wide range of different programming languages and most of them were familiar to the students. Furthermore, all of them showed to have portability, supporting different platforms and database. Given these points, we can conclude that only those factors were not determinant to choose between frameworks. So, to exclude some of the frameworks, we found that two of them were paid, and also Xojo had a difficult way to provide database support, so we excluded them. For choosing between the remaining, we searched for how to use the frameworks and some examples of code implementation for each one of them

and ordered the frameworks according to our perception of how easy they were to use compared to each other.

In the end, we decided to choose Electron framework among the remaining, since it was one of the easiest ones to use and mostly because it would allow the cohesion between languages used for the Development and Execution Environment, since the implementation based on the Electron framework is made by using the most recent languages for building graphical content, which will be useful for the web development. Moreover, this framework would allow to build applications that would be supported in many operative systems. Furthermore, some of the applications built on Electron, such as Slack or Atom, are used and known by the students and present an appealing and simple design. More reasons for choosing this framework will be detailed along the chapter.

4.2 Desktop Application

The Epik Desktop Application has as main function allow the development and exportation of the Epik games, as well as the management of all projects, activities and resources. While in the application, in order to process user requests in terms of creation, edition and deletion of the referred elements, stored in SQLite3 databases, a node.js asynchronous driver for SQLite3 is used. Written in JavaScript, the driver will provides connection/-query from SQLite3 [25].

The Desktop Application architecture is presented in Figure 4.1 as a three layer architecture. The Presentation layer is responsible for generating the content to be sent to the client, that is, what is presented to the user in the application. The displayed content will be influenced based on the processing of data at the Logic Layer, which is responsible for processing user requests according to data contained at a database or in the user's file system, depending on the request type. The database is accessible through the Data Layer.

While developing the games at the Desktop Application, the developers may perform several requests which can be in order to obtain, create or update data. Based on the request, the application will process it, communicating between the existing layers. In order to describe these behaviours, some events that may occur during the games development at the application, related to the Interactive Puzzle activities, are presented below, in the form of activity diagrams.

- **Install Plug-ins**: When processing the plug-ins installation, the application will receive the folder provided by the user and place it in a path from the user's file system were there is a specific folder from the Desktop Application destined to receive the type of plug-in to be installed. After that, the Development Environment for the type of plug-in installed will be available, allowing the user to create activities from the type the plug-in refers to (Figure 4.2).

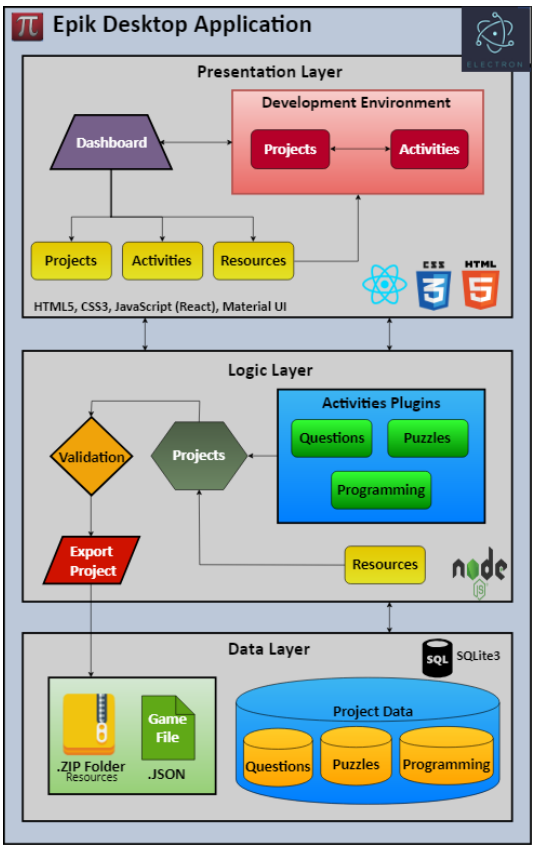

Figure 4.1: Desktop Application Architecture Scheme.

- **Add Activity to Scenario**: When the application processes the list of activities by searching for it in the database, in case there are no results, the creation of an activity is first needed in order to add it to a scenario later. In that case, the application will communicate with the database to first verify the activity plug-ins installed and allow the display of the activities creation menu. The name and type of the activity will be sent by the user and the activity will be created and stored in the database. Having at least an activity in the list, the user may select it to be included in the scenario and the application will process the request. The resultant changes will be displayed at interface level (Figure 4.3).

4.2.1 Presentation Layer

Starting with the Presentation Layer, its composition regards three main components: the Dashboard, the Project Builder, which is the Development Environment for the projects, and the Puzzle Builder, which will lead to the Development Environments for the Interactive Puzzle activities. Those are the three main interface areas for the user at the Desktop Application. From the Dashboard, users may navigate to the Project Builder by selecting a project, and in there they may include resources and, in case the Puzzle

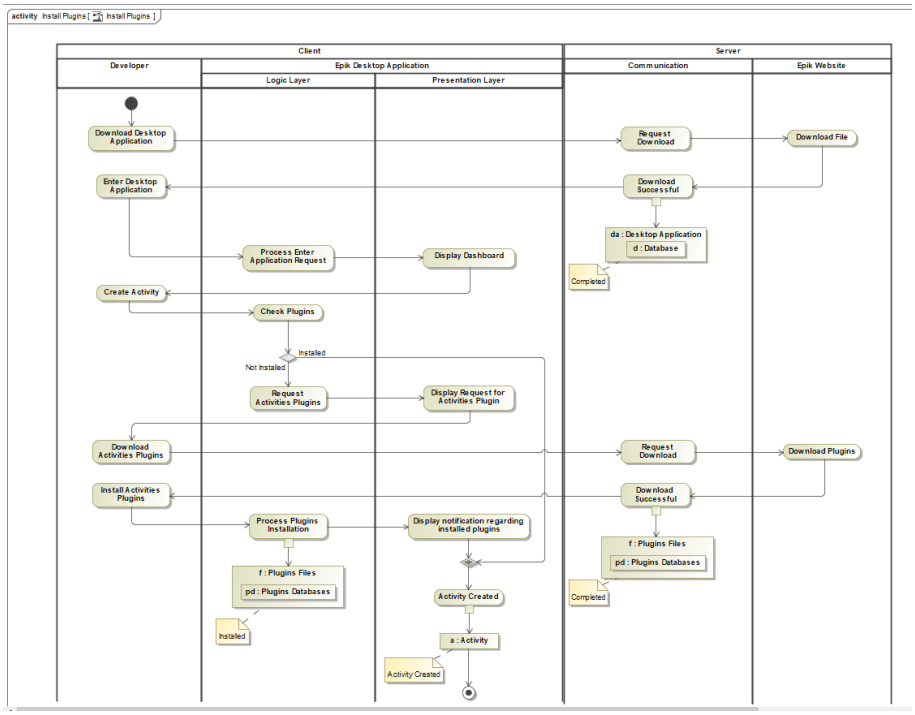

Figure 4.2: Install Plug-ins.

activities plug-in is installed in the application, activities to the scenarios. With this plug-in installed, by selecting a puzzle activity at the Dashboard or the option to enter the Puzzle Builder at activities properties when selecting a puzzle activity from a scenario at the Project Builder, users navigate to the Puzzle builder. From there, users may select a puzzle sub-type, which will lead them to the Development environment of the selected sub-type.

All the layers were built based on the selected framework, Electron, which allows to build the cross platform Desktop Application using the most recent technologies for structuring and designing graphical content [26]. With that in mind, for the design and content of the presentation layer areas referred previously, the used technologies were React, a JavaScript library for building user interfaces [27], including HTML 5 [28], CSS3 [29] and Material UI [30] (React UI Framework) components. Furthermore, Electron allows the integration of some tools, such as electron-builder, which provides a way to package and build a ready for distribution Electron application for the most used operative systems [31]. In the case of Epik, the application was distributed for Windows, MacOS and Linux. However, since some features that are compatible for Windows and MacOS are not suitable for Linux, making the Linux version incomplete with the lack of some features. More in depth, in order to move the elements inside the game scenarios, a JavaScript module for drag and drop, resizing and multi-touch gestures, called Interact.js, was used.

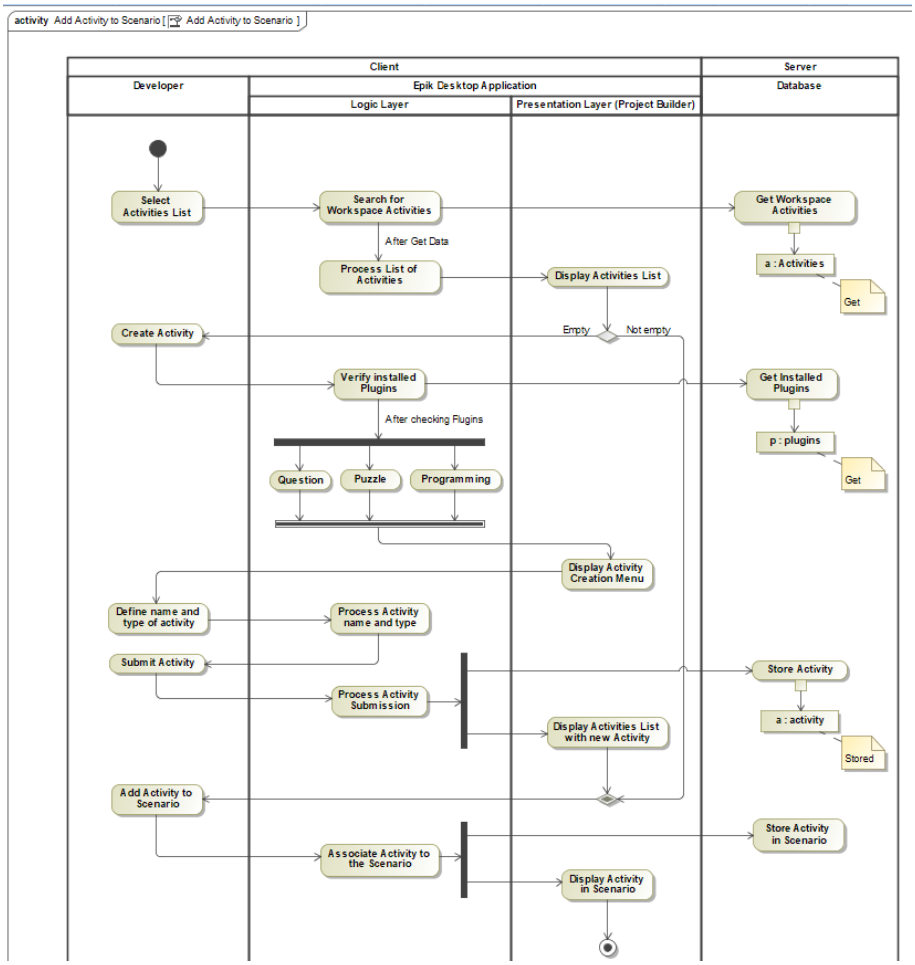

Figure 4.3: Add Activity to Scenario.

4.2.2 Logic Layer

With regards to the Logic layer, all the received requests from users are processed in the context of the resources, projects and activities, in case activity plug-ins are installed in the application. Each one of these components manages the related data according to the type of content they refer to. The component which allows the project exportation is responsible for converting the project data from the data layer to the format to send to the client, a .ZIP folder containing a JSON file with game information and a folder with the resources files associated to the game. The last described component will only take action in case the validation component approves the project in sight, according to the data retrieved by the remaining components apart from the exportation one.

4.2.3 Data Layer

Lastly, the data layer is composed by a SQLite3 Database, containing all data related to the project and application configurations in general, and for each installed plug-in, a specific database will be added according to the type of activity. To notice that information

about activities in the scenario will be kept at the project database, but specific activity details edited at their own Development Environment will be stored at the respective type database. As told previously, the data contained in this layer is accessed by the Logic Layer to be processed.

When a project is exported as a game, the logic layer will access the project data present in the Data Layer. If the generated game is valid, the logic layer will process the creation of a new game at the database, and the data related to it will be stored and structured in a .ZIP folder. The .ZIP folder will include all the resources files related to the game content, and a JSON file, containing all the data information of the generated game. The .ZIP folder is stored at a selected path of the user's operative system. With that said, the next sections will describe the selection and model of the database related to the Desktop Application, and the game file structure and organization resultant from project exportation, respectively.

4.2.3.1 Database Selection and Model

Having in mind the decisions made in terms of changes to the previous version of the platform, the data model needed to be thought accordingly to what needed to support each part. The decision for the system to use for management of data, SQLite3, relied on the simplicity to use and compatibility with the node.js driver for the communication with queries when processing user requests. After reaching an agreement among all students involved in the development of the Desktop Application, the resultant data model is displayed on Figure 4.4.

Regarding game development at the Desktop Application, the user will always have available a default workspace where they will manage all of their projects, activities, resources and games. Each project will have its general properties, including the minimum and maximum number of players, the players area and logo position, and other customizable options that need to be stored. Moreover, each project may also have collaboration and custom sounds included, but must always have scenarios, including the general scenarios described in the previous sections. Following this line of thought, each scenario may have activities and resources associated, as well as transitions, which may have none, two or three knowledge divisions associated, in case activities are included in scenarios. Both activities and resources may be of different types, but each one of them only has one type associated. Activities also can only have one sub-type according to the type associated. Moreover both activities and resources are independent from each other, meaning one can be created without the other, and will belong to the current workspace.

Regarding resources, they can be included in any scenario, where the type of resource, main characteristics and custom styles will be stored at Scenario Resource. Furthermore, the resources can also be used as a Context Resource or an Hint Resource for the activities of a certain scenario. Additionally, each audio type of resources can be used as a custom sound as well. The activities can also be included in the scenarios, where, similar to the

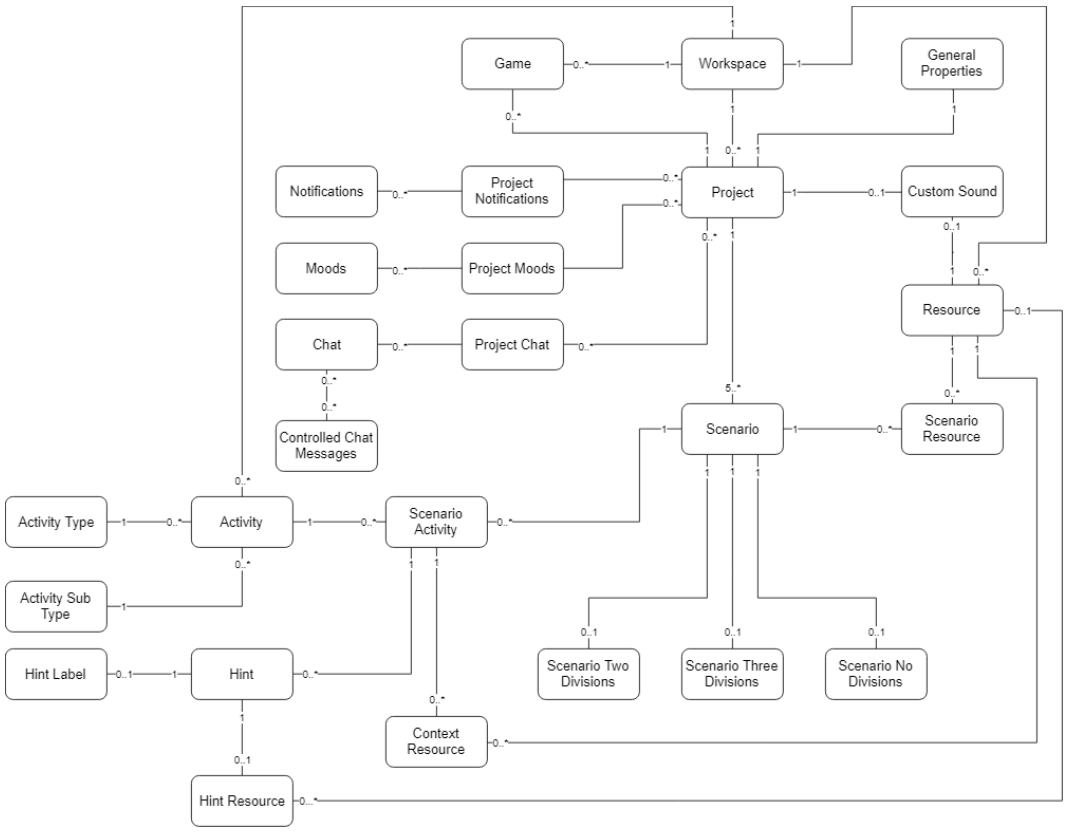

Figure 4.4: Data Model Scheme for the Desktop Application.

resources, their properties regarding the scenario will be stored at Scenario Activities. However, each activity will have its own database where properties regarding the activity content itself will be stored, taking into account its type and subtype. All things considered, specifying puzzle activities, their data model is described in more detailed in section 4.4.

4.2.3.2 Game File Structure and Organization

After exporting a project as a game successfully, a JSON file with the game content will be created. This file will store the project information, regarding mainly its general properties and scenarios content. To gather a better notion of the structure and organization of this file, at Figure 4.5 is represented a version of this file structure organized in a node tree.

As shown in the figure, there is a main node, the project, which contains all the information spread through other nodes organized in a similar way. Nodes with '@ attributes' as content contain properties such as id, background color, position and other parameters which are not relevant to display. Having that in mind, the file nodes include:

- **General Properties**: Stores general properties of the project, regarding logo and players section position in scenarios, minimum and maximum number of players,

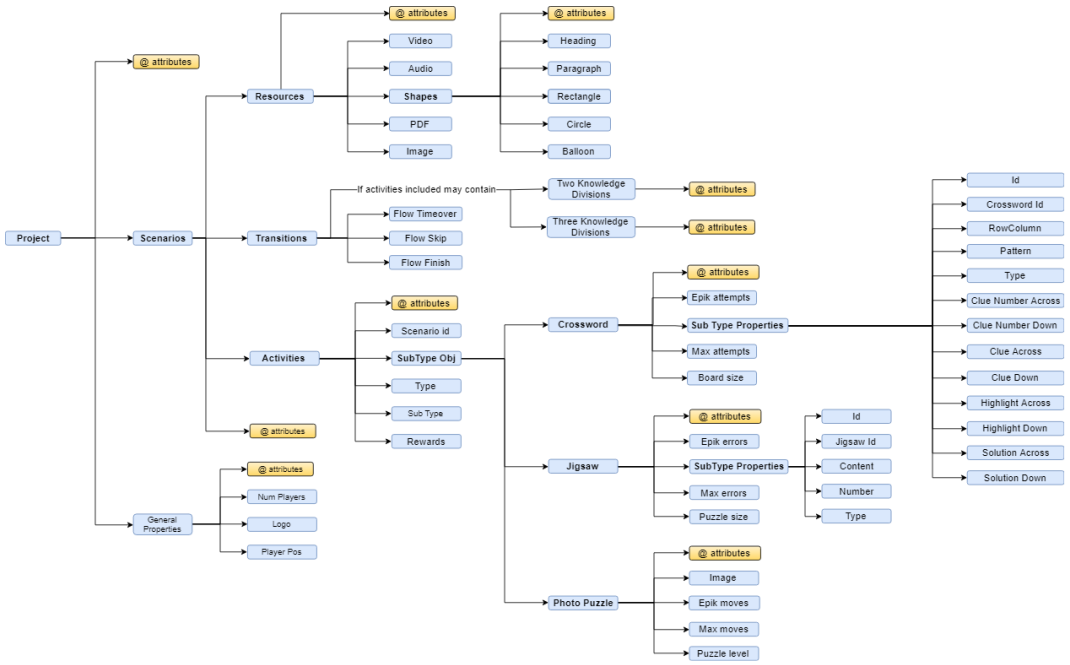

Figure 4.5: Node Tree representation of the JSON File Structure containing the Game Information.

in case the game mode is Multiplayer, player cards custom text regarding rewards, penalties, helps and total score, and other style related properties.

- **Scenarios**: Stores all the data related to the project scenarios. Each scenario will have the associated resources, transitions and activities, as well as its style and other design properties.

- **Resources**: Stores all the resources included in the project scenarios. Each one will have an unique id, name, size, position and other style properties, as well as a path to the correspondent file at the created resources folder in the generated game folder. The different shapes and their properties, similar to the resources ones, are also stored in the Shapes node contained in the resources node.

- **Transitions**: Stores all the scenarios transitions in game, according to the type of transition and scenarios ids. In case the project contained activities in their scenarios, knowledge divisions may also have been defined for specific transitions between scenarios.

- **Activities**: Stores all data related to activities inserted in project scenarios. For all activities its name, type, subtype, reward, icon position and size in scenarios are stored. Regarding puzzle activities, penalties are not defined in the game file once they are calculated in game execution. Furthermore, the 'SubTypeObj' node will always contain the activity id to what the sub type refers to, and according to the subtype of the puzzle activity, it will have specific content:

- **Crossword**: Stores the crossword board size for building the grid, the limit of attempts for having 100% of the reward and until the game is over. The content of the game, regarding the list of clues and crossword board solutions, will be store at the SubType Properties node. Each object of this node will contain the id of the crossword puzzle to which the content refers to, as well as the row and column to what the object is referring to and the associated type, where "white"targets for white squares and "black"for black squares. In case the target is a white square, the node will have stored a value for the pattern, which is the correct letter for that square, for the clue number and text to which it refers at the clue list, respective solution or solutions that use the square content and highlight on or off. For each of the last three properties referred, there is a parameter for across and down values, taking into account if the square is related to only an across, down, or both strands. On the other hand, if the target of the object is a black square, the related properties will all have null values.
- **Jigsaw**: Similar to the crossword, stores the jigsaw size, the limit of errors for having 100% of the reward and until the game is over. Moreover, the data information about the game will be store at the SubType Properties node, including the id of the jigsaw puzzle to which the data information refers to, the content and number of the object taking into account its type. If the type is 0, the object refers to a dropzone, whereas type 1 refers to a piece.
- **Photo Puzzle**: Stores the puzzle size, which refers to the number of divisions in the grid the image will have, the limit of moves for completing the puzzle with 100% of the defined reward and until the game is over, and the image path, whose file is located at the resources folder created at project exportation. Once this puzzle doesn't have individual objects, there are no SubType Properties associated.

4.3 Execution Environment

The Execution Environment is responsible for managing all the developers information and respective games they import to the server, as well as all the game sessions and players information during the game. To implement it was used the Electron framework, which includes NodeJS which allows the implementation of servers which support many types of protocols, such as the WebSocket [32], which allows the communication between client and server without having to re-establish a channel per request. Furthermore, Electron provides a way to use the same programming languages to develop both client and server from the Execution Environment, as well as the Desktop Application. Together with this framework was also used the Socket.io library [33], which helps in the process of managing the communication channel between server and client.

With regards to the architecture of the server, its is built based upon the architectural pattern MVC, which divides the application in three main components: Data Model (Model), execution and processing controller (Controller) and data visualization (View) [34]. Having that in mind, the architecture is presented in Figure 4.6, where the communication layer is responsible for processing user requests, which are processed in the logic layer which may have to access the database, present in the Data layer, depending on the request. After the request is fulfilled, the communication layer will distributed the results to the presentation layer, where they are displayed to the client.

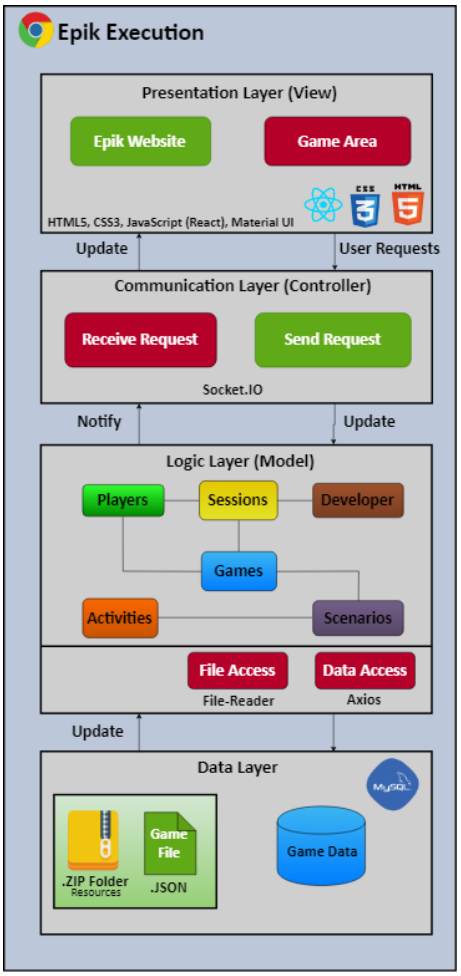

Figure 4.6: Execution Environment Architecture Scheme.

The Execution Environment behaviour when importing a game is presented below, in the form of an activity diagram 4.7. The controller will process the unzip of the game folder and the storage of the resource files and the game file in the server file system. The resource files will have the project name included in their name in order to identify to which project each resource belongs to. Moreover, some game information from the game file will be stored at the server database as well, which later on can be used to generate the session logs. After verifying that the inserted date of availability of the game is valid, player emails can be associated to the game and the controller will process their storage

in the server database. When all is done, the game will be ready to be submitted to the Execution Environment.

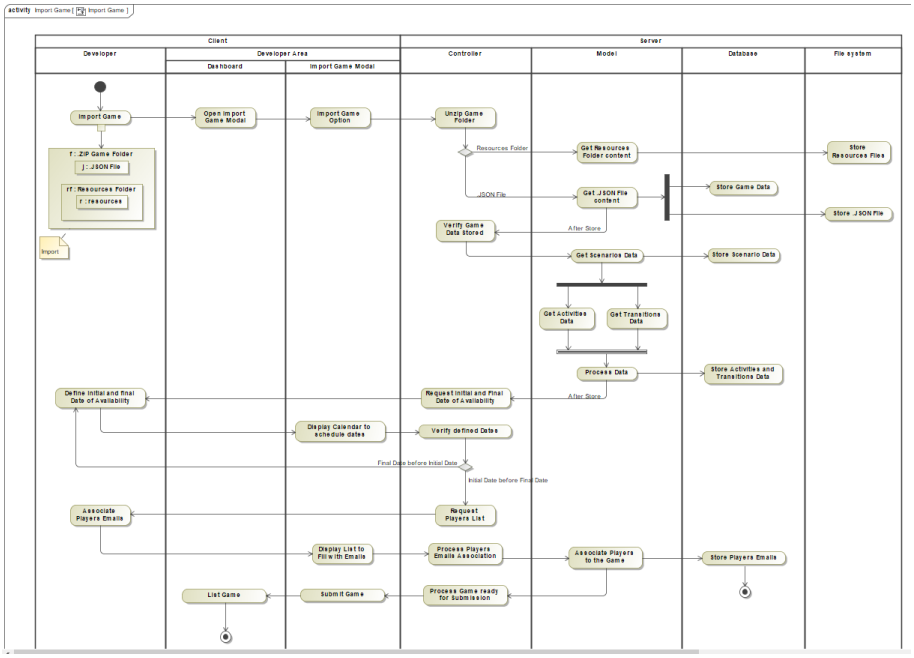

Figure 4.7: Import Game.

4.3.1 Presentation Layer

The presentation layer, as told previously, is where the client can interact and visualize data. As Electron framework was used to build the Execution Environment, the languages used to structure and design the presentation layer were the same ones that contributed to the development of the Desktop Application. This layer is divided in two components, the Epik Website an the game area. The Epik Website is where the download area for obtaining the Desktop Application along with the activities plug-ins is located, as well as informations regarding the Epik platform context. Moreover, this area allows to access the Game Area, which is divided in another two areas, the Developer Area and the Player Area. At the Developer Area it is possible to request the importation of games, access to session logs and edition of game and account information, as well as playing own games and games which the user is assigned to. At the Player area its is only possible to request the execution of games the user was assigned for.

4.3.2 Logic Layer

The logic layer contains several components which, similar to the logic layer from the Desktop Application, will be responsible for processing the users requests, along with the Communication Layer, managing the data referent to their context. With that in mind, the user requests will always come from the players, since they are the ones interacting

with the interface, that is, with the presentation layer. Therefore, through a player request it is possible to reach the session he is in and respective game. By recognizing the game, it is possible to communicate with the scenario or even with the activity in the scenario, depending on the location of the request. In fact, each of these components might have to communicate with the data present in the data layer in order to process some requests. Processes regarding game information will be performed by the file access component, which uses FileReader, a module which lets web applications asynchronously read the contents of files [35], in order to obtain data related to the game. On the other hand, processes might need to use the data access component, which uses Axios, a HTTP client for the browser and node.js that allows to make HTTP requests from node.js, to get data to be included in the game or store data related to session logs [36].

4.3.3 Data Layer

Taking into account the generated game folder from the Desktop Application, most of the game information is obtained directly from the JSON file and processed to the presentation layer without needing access to the database. Therefore, the construction of the database model for the Execution Environment became a lot simpler. MySQL was the select system to manage data at the Execution Environment, once it is simple to use and all students were comfortable using it since they had already previously worked with it. On the whole, the resultant data model for the Execution Environment is represented in Figure 4.8.

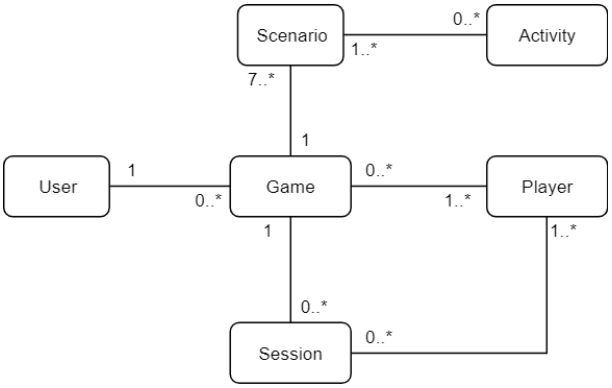

Figure 4.8: Data Model Scheme for the Execution Environment.

Based on the reasons presented in the previous paragraph for the characteristics of the database model for the Execution Environment, its main objective is to store information to be gathered and retrieved to the developers as game session logs. Therefore, information regarding the developers, represented as 'User' in the model, must be stored in order to gather information about the games imported to the server and who imported them. Regarding the players assigned to play the games, the only relevant data to store is their email in order to identify them in the sessions they played. Furthermore, each time a game is played a session is created, associating all the players that participated in the

session and storing data about their performance regarding the activities and scenarios played, in order to gather information about the game sessions to be presented to the developers as logs. These informations stored also contribute in building both individual and team rankings.

4.4 Interactive Puzzle Activities Implementation

In terms of the Interactive Puzzle activities, their goal is to extend the set of activities available for the Epik Platform, in a way that would contribute to it as an educational activity. For each type of puzzle included, at the Desktop Application, a specific Development Environment was created. These Development Environments were built using the same tools as for the Desktop Application, in order to keep the coherence and consistence of the platform. However, they are only accessible if the Interactive Puzzle activities plug-in is installed at the Desktop Application. For the case when this plug-in is not installed, an import plug-in folder feature, which is the install function, was implemented (Figure 3.9). If the user tries to create a puzzle activity without the plug-in, a message will appear demanding its installation. The installation process is demonstrated in Figure 4.2 and explained in the text section referring the diagram. The Application will recognize the plug-in installation since when the plug-in folder is placed in the Desktop Application folder at the user's file system, the database is updated and recognizes the presence of the plug-in. However, it is necessary that the user restarts the application for the changes to take place.

The Interactive Puzzle activities and their Development Environments at the Desktop Application are included in the Desktop Application architecture shown in Figure 4.1 and described in section 4.2. Their presentation layer only includes the Development Environment of each type of puzzle activity and the Puzzle Builder main page (Figure 3.37), while the puzzle sub-type is not defined. When a sub-type is selected at the Puzzle Builder, the logic layer will communicate with the data layer and update the sub-type value of the current activity. Thus, the next time this activity is accessed, according to the sub-type selected, the associated Development Environment will be displayed instead of the Puzzle Builder. All things considered, for each type of puzzle, additional tools beyond the Electron framework were used in order to achieve the final results:

- **Photo Puzzle**: An integrable tool with Electron, called react-image-puzzle, was used for creating a swappable tile puzzle out of an image [37]. The same tool includes an editable feature to know when all the pieces are in the correct place to form the final image, which allows to customize the events to trigger in the end of the puzzle. Additionally, an event trigger was created each time a piece was moved, in order to know the amount of moves performed by the user (Figure 3.43).
- **Jigsaw Puzzle**: The movement of the pieces and drop mechanics were achieved using Interact.js module for drag and drop elements. The same mechanism properties

were changed in order to know when a piece matches a certain dropzone, which happens when a piece is dropped in a dropzone and they have the same number associated. This will trigger an event which counts each time a piece is not in the correct location and changes the piece color to green or red, according to whether there was a match or not, respectively (Figure 3.44).

- **Crossword Puzzle**: A CSS and HTML working crossword puzzle static example [38] was reimplemented using React in order to allow its construction from ground zero and full customization, regarding all verifications to prevent users from building crossword puzzles with mistakes and checking correct word insertion in game (Figure 3.40).

As for the scoring algorithm, all the puzzle types start with a score percentage of 100%. According to the defined limit for the puzzle to be completed and keep the total reward defined for the activity, that is, 100% of the score, if it is not exceeded, the player will receive the total points corresponding to the reward attributed to the activity. For each time the limit is exceeded, a score % will be lost according to the limit defined for having 100% of the score and for losing the activity. When the score percentage reaches 0%, that means the limit defined for the activity to end was reached aswell, and the activity ends, returning no rewards for the players. Besides, when an activity is completed, the penalty % regarding the player performance is removed from the total reward of the activity, and the resulting reward is added to the player score in the game.

With regards to the Puzzle activities Database model, Figure 4.9 displays a representation of it at the Desktop Application.

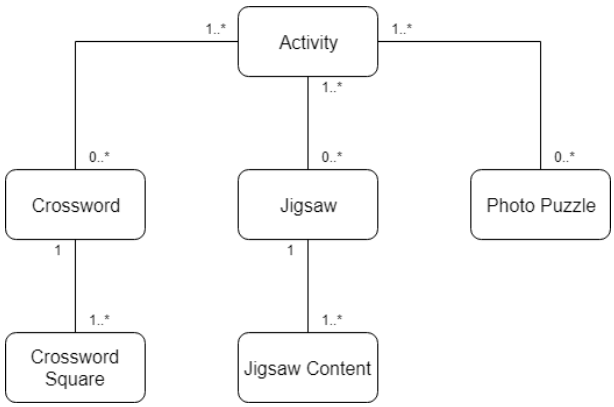

Figure 4.9: Data Model Scheme for the Interactive Puzzle Activities.

Taking into account the database model from Figure 4.4, when creating an activity, despite being required to define its type before submitting its creation, the sub-type can be edited later. Therefore, there is the possibility to have activities without a sub-type associated. However, if these activities are included in a scenario, the game will always be invalid since one of the verifications made for project validation is that activities must have a sub-type and be valid at their own Development Environment. Furthermore, when

the activity is of puzzle type, there is the possibility to choose among three types of puzzle, which are the sub-types: Crossword, Jigsaw and Photo Puzzle. When a Crossword puzzle is created, the associated crossword squares, according to the defined board size, are also generated. Each crossword square will store information regarding if they contain a letter or a black square, their row and column, the solution or solutions and clues associated, in case they contain a letter, and other minor properties. The same situation occurs in the case of the Jigsaw Puzzle, where depending on the puzzle size, the jigsaw content, regarding pieces and dropzones, is created, storing information on the content and number of each piece and dropzone, as represented in database model from Figure 4.9.

As told previously, before being able to include the activities in the game scenarios, validations had to be implemented in order to avoid having incomplete or incorrect puzzle activities. Therefore, for the Photo puzzle to be valid the logic layer will only verify if the photo field containing the image path is different from the default one in the data layer. All the pieces and dropzone area on the Jigsaw Content table must have their content edited for the Jigsaw puzzle to be valid. Similarly, in order to have a valid Crossword puzzle, all the squares of the board stored at the Crossword Square table must be filled whether with a letter or a black square. Validation is checked everytime activities changes are saved at their own Development Environment, meaning that the logic layer will communicate with the data layer to see if the puzzle stored data contains the requirements for being valid. When not valid, a notification will be sent to the presentation layer showing which of the validation requirements is missing.

Furthermore, for both Jigsaw and Crossword puzzles, other validations were implemented in their development stage at the Desktop Application when inserting content, to avoid mistakes. Besides checking if a piece and dropzone have content inserted, the Jigsaw content has a limit of characters so that it doesn't surpass the limit of the piece or dropzone. Additionally, the number both piece and dropzone share in order to see if they match must be filled in. In case a number which already belongs to a certain set of piece and dropzone is selected, their content will be edited and no new set is created (Figure 3.45). Regarding the Crossword puzzle, follows a list with the implemented errors (Figure 3.39) to avoid when inserting content at its development stage:

- Overlap content with the same orientation on the board, except in case of edition;
- Adding content oriented down when there is already content oriented across and content oriented down with the same number inserted;
- Adding content oriented down when there is already content oriented across with the same number inserted, except in the initial letter;
- Adding content oriented across when there is already content oriented down and content oriented across with the same number inserted;

- Adding content oriented across when there is already content oriented down with the same number inserted, except in the initial letter;
- Overlap content in the same orientation on the board with a different initial letter in case of intersection with the initial position, except in case of edition;
- Overlap new content over a black square.

As for the exported content in the game file regarding puzzle activities, a detailed description of it can be found at section 4.2.3.2. In the Execution Environment, there is no specific database for the puzzle activities, since, as explained in section 4.3.3, all game data is obtain from the game file, and the only data to be stored related to puzzle activities are its type to distinguish it from other activity types when returning logs to the developer.

Moreover, all the developed puzzles were already playable after being added content, for testing purposes in the Desktop Application. Therefore, as told before, their adaptation to the Execution Environment was simplified. An exception to this were the Crossword puzzles, where the solutions appeared on the board already completed in the application so that the developers could see the content they were adding. Therefore, at the game execution, the board appears unfilled and a method was implemented which verifies if the letter the player types in a certain square is equal to the one obtained from the game file information. For each word inserted, the square color changes to green if the word is correct and red when incorrect. When incorrect, penalties can be associated, according to the rules from the game file, as exemplified in Figure 3.57.

4.5 Chapter Summary

Throughout this chapter is presented the implementation options and decisions taken regarding the Epik platform and the Interactive Puzzle activities. It primarily contains the choices regarding the framework selection for developing the Desktop Application in concordance to some requested characteristics, which also allowed to use components that were used for building both the Execution Environment and Interactive Puzzle activities. Furthermore, the database selection and model, as well as the architecture style used for each part of the platform is explained, justifying the choices made.

Chapter 5

Platform Testing and Results Analysis

The presented chapter displays the results analysis regarding platform testing and answers given to a survey about Epik Desktop Application experience to the users, in order to evaluate Epik games development, including the development of Interactive Puzzle activities regarding the installation of their plug-ins in the application, as well as their usage in education and user satisfaction with the application. A notion of the meaning of the results obtained from the survey regarding the dissertation context is detailed, as well as the conclusions which can be taken from them.

5.1 Survey and Respondents Description

The main objective of this survey was to obtain different opinions that would allow the evaluation of different aspects regarding the development of games and Interactive Puzzle activities at the Desktop Application. Moreover, another important aspect to evaluate was how the use of games in education had evolved, if there are still some reticence or if its more commonly used, and if the Epik games could help influence in that matter.

With that in mind, the survey was divided in different sections according to the aspects to evaluate. The first sections regard questions related to the tester and their opinion on games as a learning tool, and the remaining ones are directed to the Desktop Application and their different areas, including the Interactive Puzzle activities Development Environment. Most questions asked the respondents to check one or more answers from a list or to evaluate on a scale from 1 to 5, representing negative and positive, respectively. The intermediate value 3 represented indifference or neutrality to the aspect being evaluated. Lastly, at the end of the survey there is a section where respondents could leave comments, critics and suggestions on the overall testing.

Furthermore, as the development of Epik games has the objective to be further used as a learning tool for players, the target respondents for this part of the platform and associated survey were teachers. Therefore, a group of 21 teachers was asked to perform some tasks at the Epik Desktop Application and then answer to the survey presented at section I. From the group, approximately 67% were females and 33% were males, between the ages of 20 and 60, with some years of experience as teachers, as shown by Figure 5.1. Additionally, has shown in Figure 5.2 the educational grades taught by the respondents was varied, which was a good factor in order to have different perspectives about the aspects to evaluate.

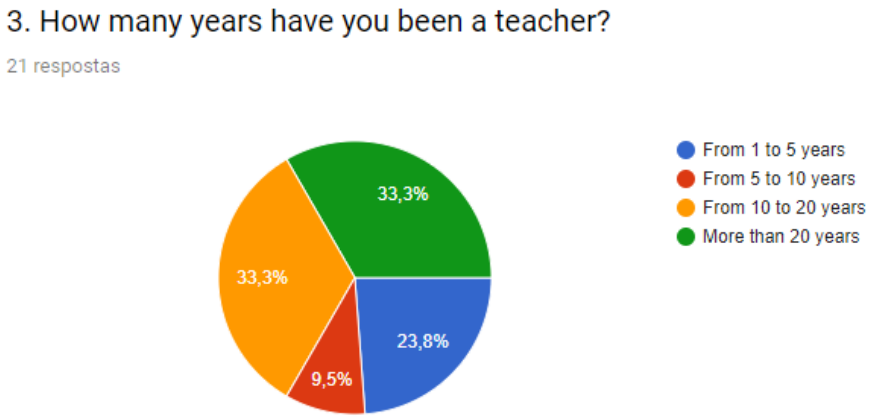

Figure 5.1: Years as a Teacher of the Respondents.

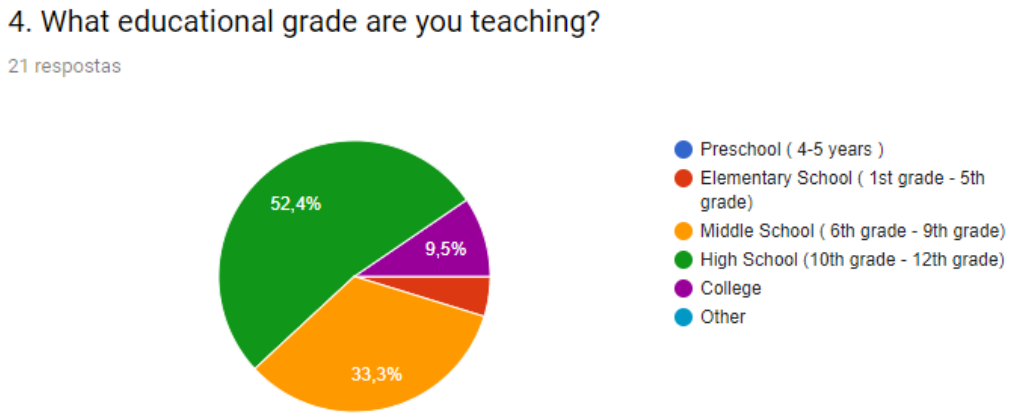

Figure 5.2: Educational Grades taught by the Respondents.

The results analysis regarding the Interactive Puzzle Activities Development will be discussed in the next section. The remaining topics of the survey, such as the use of games and, in specific, Epik Games as a learning tool in education will be presented at section 5.4 and the ones related to the Epik Desktop Application at section 5.3.

5.2 Interactive Puzzle Activities Development

When questioned about the development of Interactive Puzzle Activities at the Epik Desktop Application, overall, the group of respondents opinions were very positive. On the whole, respondents found the Development Environment of these activities easy to use (Figure 5.3), and that its inclusion in the Epik games would be of a great use, having 42.9% of the respondents rating this aspect 5 out of 5 (Figure 5.4).

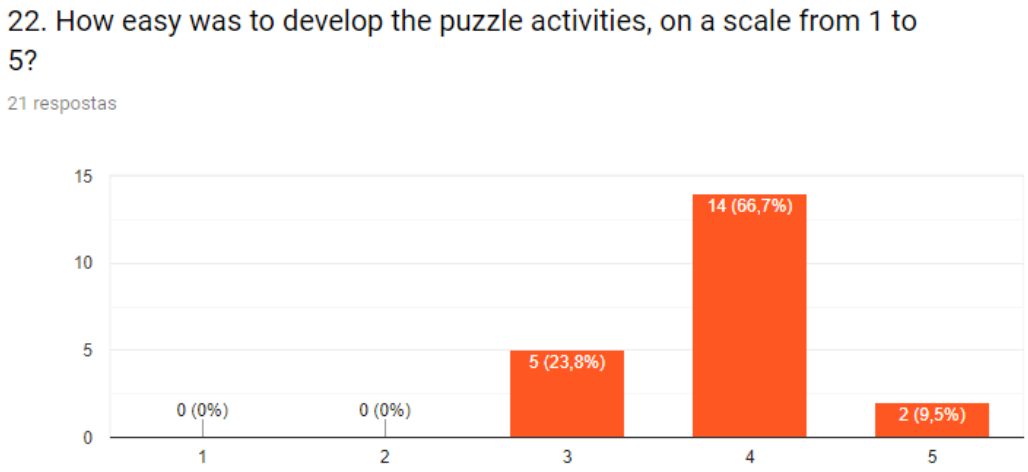

Figure 5.3: Respondents Opinion on the Ease of Use of the Interactive Puzzle Activities Development Environment.

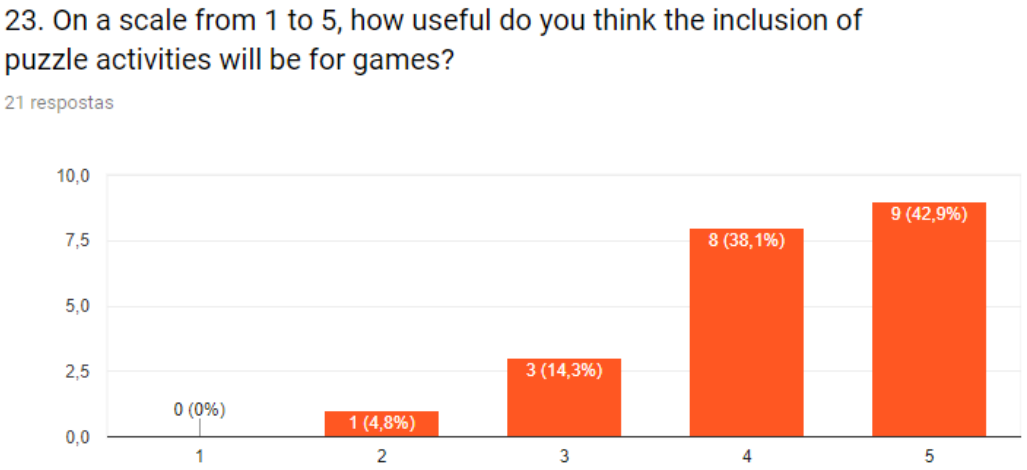

Figure 5.4: Respondents Opinion on the Inclusion of Puzzle Activities in the Epik Games.

Also related to the last aspect evaluated, the diversity of educational subjects that could be included in the context of the game with the use of activities of the puzzle type also had great results, having 61.9% of the respondents rating this aspect at least 4 out of 5, as shown in Figure 5.5. The fact that both last referred aspects had positive reviews is very important since it shows that the puzzle activities would be able to be used for

learning purposes regardless of the educational subject. Regarding individually the 3 types of puzzle available, almost all of the respondents (17) qualified the Photo Puzzle development interface user friendly, the Jigsaw Puzzle development interface opinions were more divided between user friendly and slightly confusing and the Crossword development interface was the one receiving more slightly confusing and not user friendly reviews (Figure 5.6). These results about the Development Environment of each type of puzzle activity were expected, since both Jigsaw and Crossword puzzles content can't be changed directly in their pieces and dropzone or board, respectively. There is a feature for each one of them that opens an edition menu, allowing the customization of the content, which can be slightly confusing to some users.

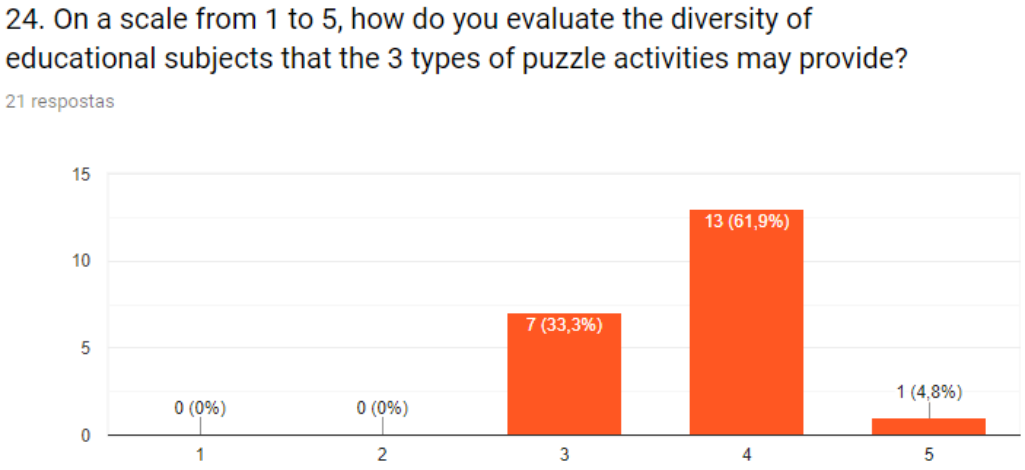

Figure 5.5: Respondents Evaluation on the Diversity of Educational Subjects provided by the Puzzle Activities.

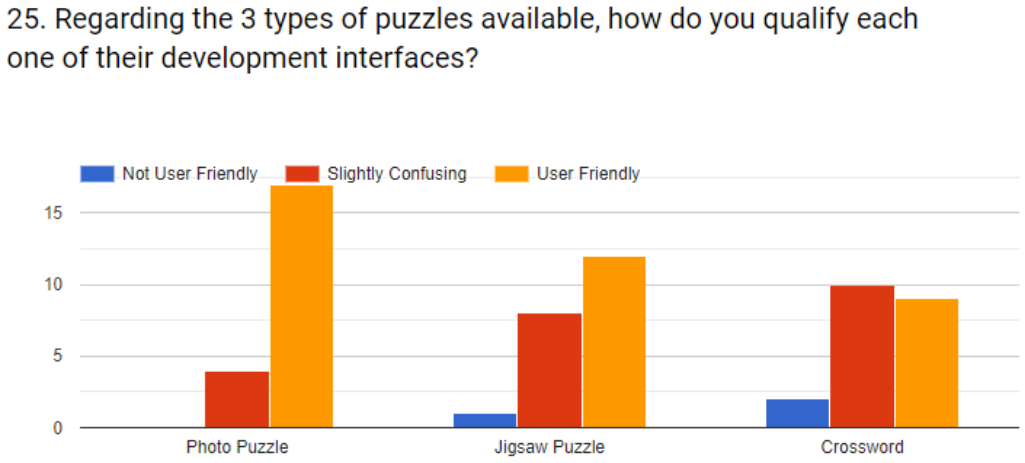

Figure 5.6: Respondents Evaluation on the Development Environment of each Type of Puzzle Activity.

5.3 Epik Games Development

When evaluating the Desktop Application interface, the majority of the group of respondents found it user friendly (question 10), shown in Figure 5.7, along with the tutorial helps provided by the Epik manual (question 11), shown in Figure 5.7,which revealed to be very useful, having 52.4% of the respondents evaluating its utility a 5 out of 5 and 38.1% rating 4 out of 5. Both parameters being evaluated that high was a very important achievement, since one of the objectives of the Desktop Application was to provide an easy way to develop educational games. However, regarding overall error description (question 14), 47.6% responded neutrally and 28.6% rated 2 out of 5, which tells that may be the errors description should have been more specific in some cases and may be more suggestions on how to solve the error should have been given.

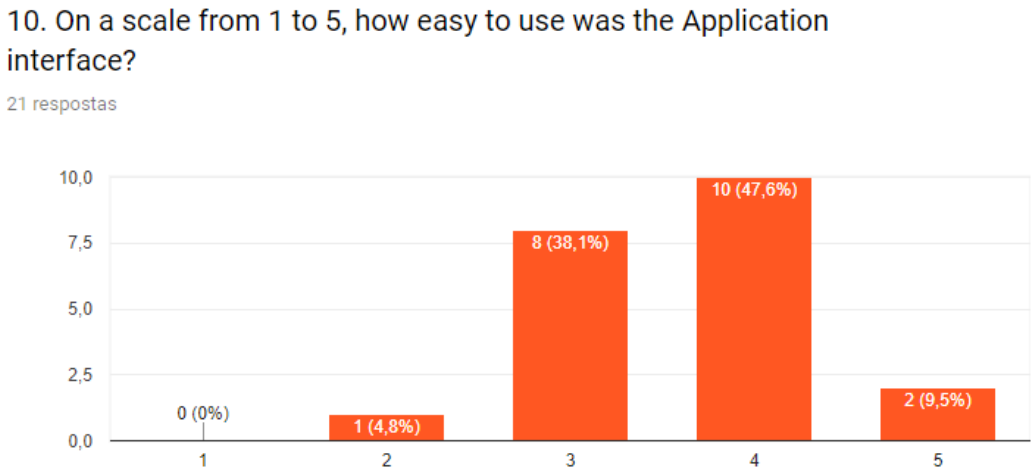

Figure 5.7: Respondents Opinion on the Application Interface Ease of Use.

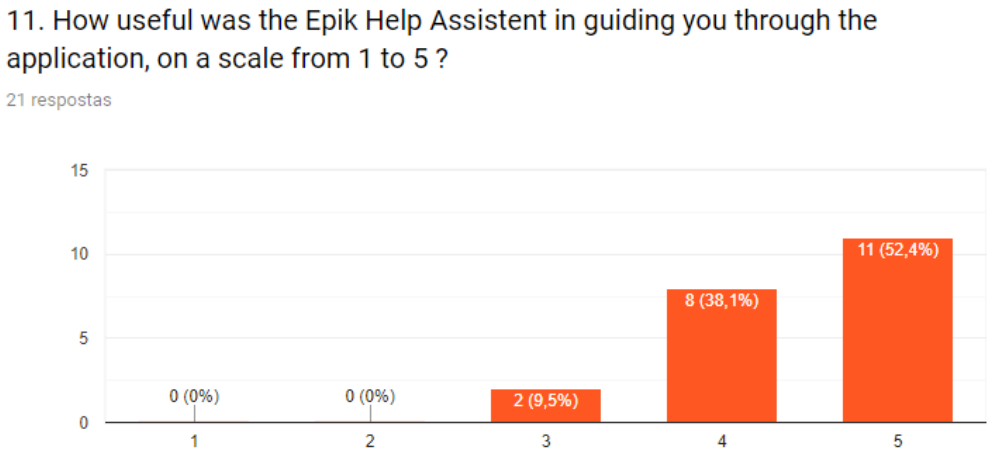

Figure 5.8: Respondents Opinion on the Application Help Assistance Manual.

Furthermore, in terms of overall interface organization, displayed in Figure 5.9, 42.9%

evaluated it a 4 out of 5 and the same percentage was obtained as a neutral opinion. According to the amount of components and elements the application interface agglomerate, its understandable that for the first times using the application it may not be easy to get familiar with all the interface details. Moreover, the game organization in scenarios (question 15) was rated 4 out of 5 by 57.1% of the respondents and neutral by the remaining ones, which can be interpreted as a need to may be have more options in the future, instead of only organizing games by scenarios.

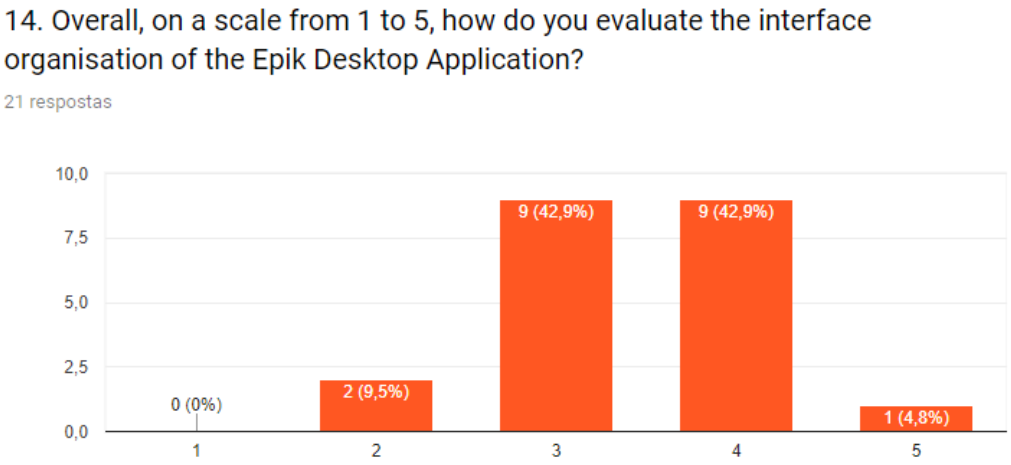

Figure 5.9: Respondents Opinion on the Organization of the Application Interface.

Taking into account the scenarios content, namely resources, its insertion in game scenarios (question 17) was considered to be very intuitive and easy to perform by approximately 10% of the respondents, rating it 5 out of 5, and 57.1% also rated it 4 out of 5. Related to this topic, the utility of the resources in games were rated 5 out of 5 by 38.1% of the respondents and 52.4% rated it 4 out of 5, as displayed in Figure 5.10. Therefore, it can be assumed that the type of resources selected to be included in games development, as well as the method to include them in game scenarios, contribute in a positive manner to the games.

5.4 Use of Games in Education

With regards to the use of games in education, the questions made in this section had as objective evaluate the opinion of the respondents on the use of games in education and their experience in this topic, as well as if the Epik platform could be used to develop games that would serve as a learning tool in education. Overall, the answers from each of the respondents of the group was positive, and will be presented along the section.

When asked about the opinion on the use of games in education, as shown in Figure 5.11, 38.1% answered that was 'very useful as a way to capture students attention', 9.5% think that it is 'Not very useful as students may get distracted', 47.6% stated it as 'Only useful if the games are well-structured' and 9.5% found it 'very useful as a way to pass the

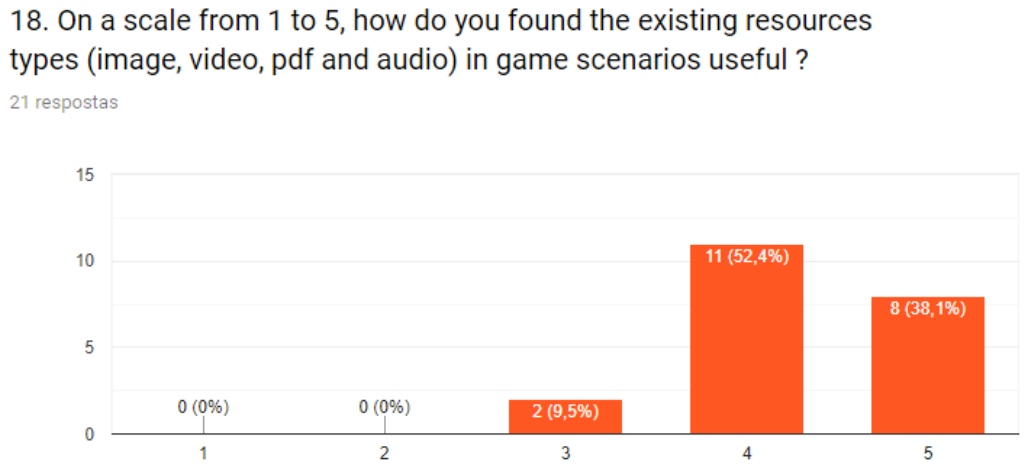

Figure 5.10: Rating given by Respondents on the Utility of Resources in Games.

knowledge to the students'. On the whole, it was a very positive result, taking into account that the majority of the respondents (62%) never used games as a learning tool for their students (question 6) and 90% never created games using applications (question 8) and still think that games are useful for education in a certain way. The 38% of respondents used games for Tutorial classes, Laboratory classes or Homework (Figure 5.12), and the 10% of the respondents that already used applications to create educational activities only used Moodle (question 9), in the form of surveys.

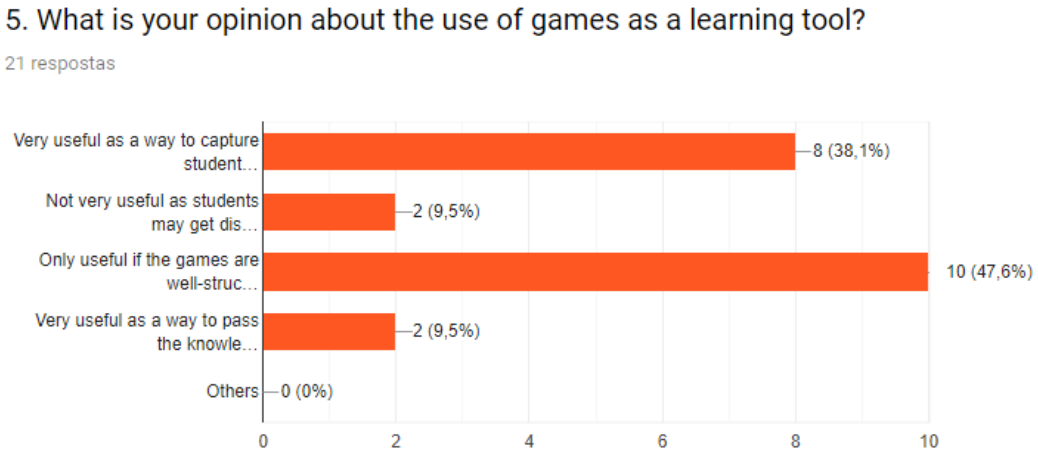

Figure 5.11: Respondents Opinion on the Use of Games as a Learning Tool.

After the sections with questions regarding the Desktop Application interface, respondents were asked if they would use the Desktop Application to develop games and use them in class as a learning tool. As shown in Figure 5.13, 71.4% of the respondents responded positively, and 70.6% of them would use the Epik games as homework for their students, as displayed in Figure 5.14. Taking into account that, as shown previously,

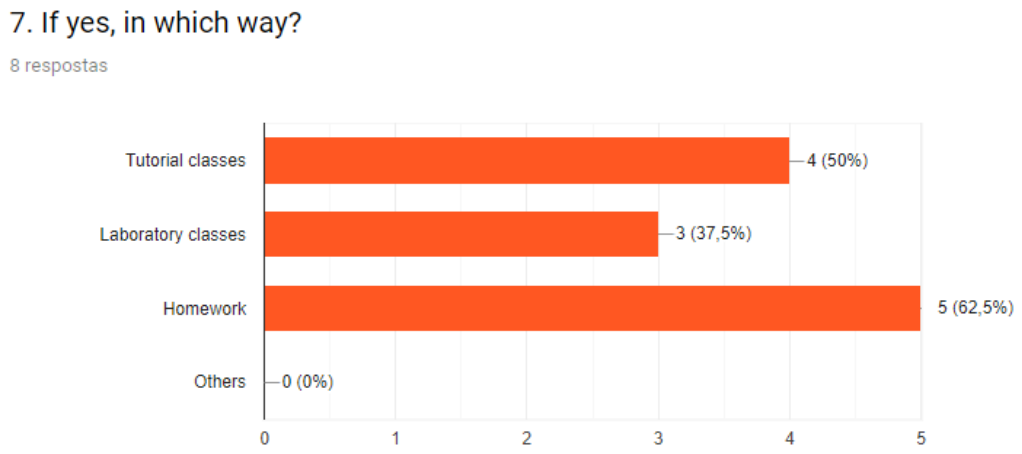

Figure 5.12: Ways of using Games for Students by the Respondents.

the majority never used an application to create educational games, the results were very positive.

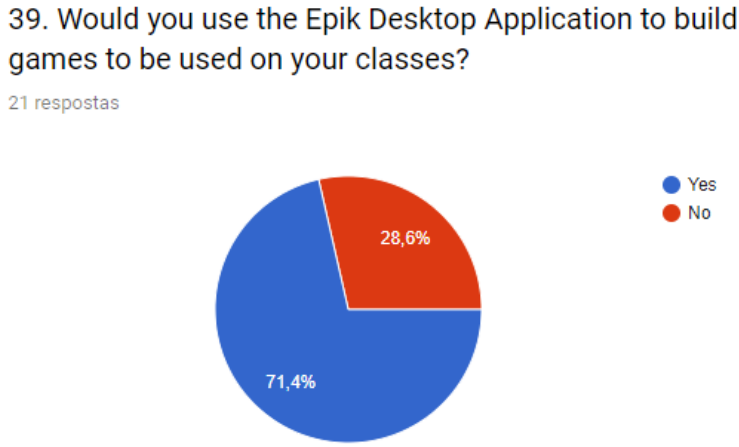

Figure 5.13: Respondents Opinion on using Epik Desktop Application for building Educational Games.

5.5 General Satisfaction with Epik Desktop Application

The last section of the survey evaluates the overall satisfaction of the respondents with the Desktop Application. In fact, as seen in Figure 5.15, 57.1% of respondents rated their satisfaction with the application a 4 out of 5 and 33.3% gave a neutral opinion. In addiction, in terms of the characterization of the Epik Desktop Application words like Friendly, Useful, Simple but also Confusing were the most used, based on Figure 5.16. Putting all together, along with 66.7% of the respondents saying they would recommend the Epik Desktop Application (question 42), the feedback show that some improvements

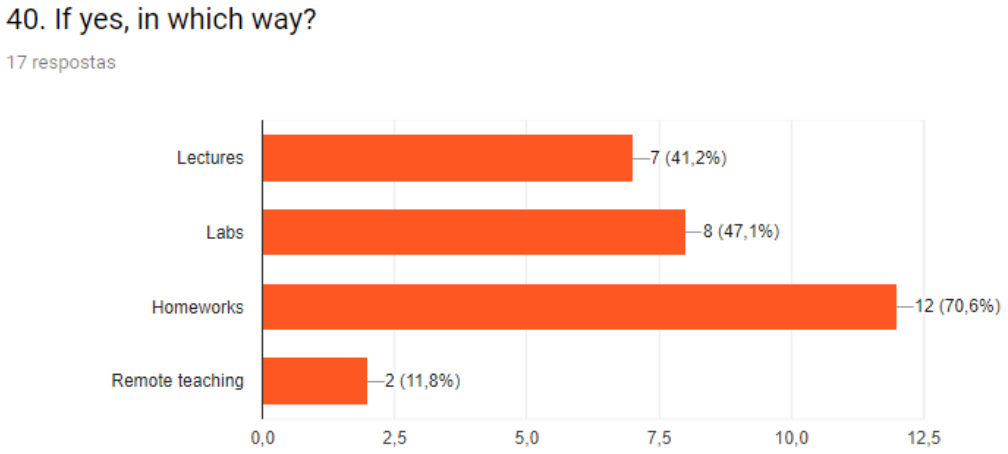

Figure 5.14: Ways of using Epik Games in Classes by the Respondents.

should be done in order to make it easier for the users to get familiar with the application, despite the majority having approved the concept.

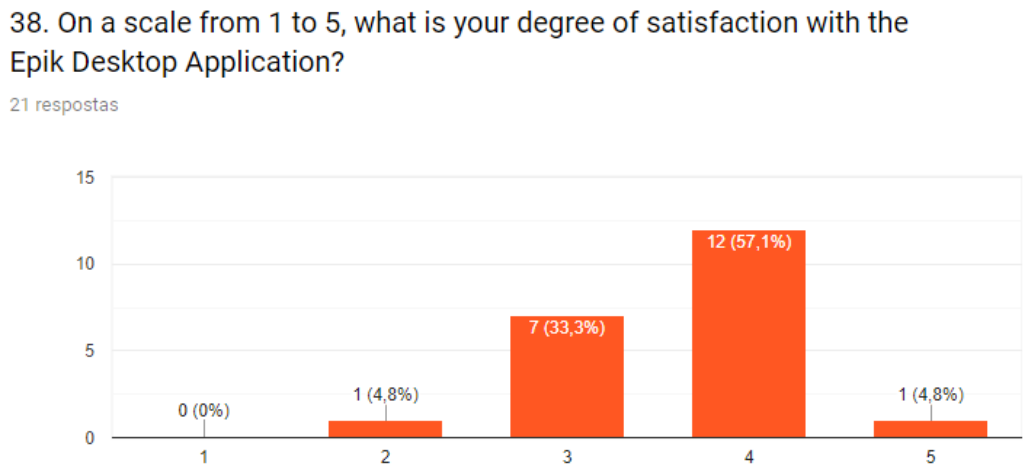

Figure 5.15: Respondents Degree of Satisfaction with the Epik Desktop Application.

Lastly, suggestions and commentaries were left by some of the respondents. Next follows a list of the most relevant ones:

- "Still with some flaws to be corrected but useful for explanations."
- "In the construction of the graph the disposal defined for the nodes should be saved".
- "Missing consistency in some error messages."
- "Some modifications are still saved even when changes are discarded."
- "The idea of having different types of activities in scenarios is very good."

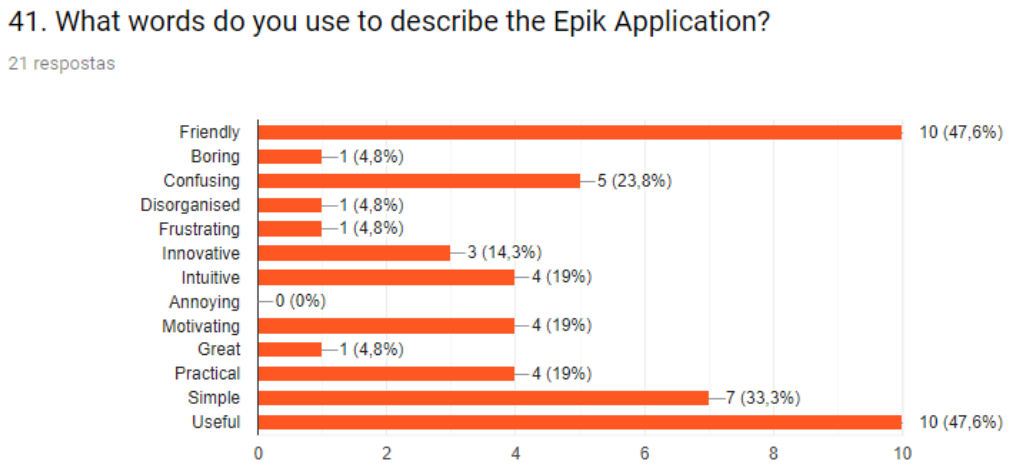

Figure 5.16: Words Respondents selected to Describe the Epik Desktop Application.

- "I only understood what a game would look like through the helps manual. Was difficult to use."

All the provided commentaries and suggestions described some aspects that should be taken in consideration for future work in order to improve the overall user experience with the application.

5.6 Conclusions

To sum up, the overall results obtained from the survey were positive and, regarding this dissertation's theme, they meet its objectives. The reviews from the Interactive Puzzle activities development were in its majority positive, regarding ease of development, diversity of educational subjects to be integrated in the game context and overall inclusion in the Epik Games. An exception was the crossword puzzle Development Environment, where some respondents found it not user friendly. All things considered, the fact that most reviews were positive demonstrate that puzzle activities might have the potential to be used for learning purposes. Apart from that, there was a considerable amount of neutral and in a few aspects even negative reviews about the overall development of games at the Desktop Application. In fact, the most negative aspect pointed to the application was being confusing, which, as referred previously, can also be due to the fact that respondents were trying the application for the first time and were not familiar with some features and terminology associated. Although some of these reviews revealed some flaws, in general, the process of developing games at the Desktop Application was considered simple and respondents showed interest in using the application. On the whole, this negative aspect and the Crossword Development Environment one, as well as the suggestions and commentaries will be taken into account as future work.

Chapter 6

Conclusions and Future Work

The presented chapter includes all the conclusions regarding the developed common and individual work at the Epik platform, enumerating some improvements and features that can be included in future versions of the platform.

6.1 Conclusions

For this dissertation was proposed to design and develop Interactive Puzzles to be included in the Epik platform, in order to extend the range of activities available in the platform and in a way that would contribute to it as an educational activity. Regarding the Development Environment for each of the selected puzzle activities and their execution in the Epik Games, overall, the objectives were accomplished. First a study was made about the existent puzzle games and how they could be adapted to the platform, regarding the compatible frameworks to build them according to the ones that would be used for the development of the overall platform. After selecting the puzzles to be developed and included in the platform, regarding framework limitations, diversity of educational subjects and ease of use in terms of development and execution for the users, the Development Environment which would allow to customize and build each one of them was created. These environments were developed in parallel with the Desktop Application, being accessible when a puzzle activity is selected on the Dashboard or in a scenario at the Project Builder. Moreover, it was necessary to create validation rules for each one of the developed puzzle activities so that developers won't provide incomplete or incorrect games to the players. After being validated, the puzzle activities could then be used in games. When the exportation of games without activities in the Desktop Application started to work correctly, the feature was improved in order to allow the inclusion of puzzle activities in the game file. Hence, the game file had to be modified and organized

in a different way to support the information from the puzzle activities.

Having accomplished these objectives, there was a need to have a plug-in system for the different types of activities, including the puzzles, which would allow the platform to be less heavy in terms of file size and give users the option to select which ones they would like to include in their games instead of having to download them all. To do so, the files from the Desktop Application which contained all the information about the Interactive Puzzle activities and their Development Environment were made available for download at the Epik Website, and a solicitation for the installation of these files was created when any operation related to them was requested at the Desktop Application. To perform the installation, the files downloaded are inserted in the correct path of the Desktop Application files in the user file system, when the install feature is selected at the Desktop Application. Overall, since all the types of puzzles were already playable after being added content, for testing purposes in the Desktop Application, their adaptation to the Execution Environment was simplified. An exception to this were the crossword puzzles, which appeared completed in the application so that the developers could see the solutions they were adding. Another fact that contributed positively to the adaptation to the Execution Environment was that the game information is obtained directly from the game file, meaning no database for each type of puzzle was required. With that in mind, when selecting the puzzle icon in a scenario containing a puzzle activity at game execution, a custom window was created which displays the puzzle activity to complete and the activity rules defined by the developer. Once terminated, the final score obtained in the activity is added to the player total score.

In terms of the common work of the dissertation, starting with the development of the Desktop Application responsible for the development of the Epik games, in an initial phase decisions had to be taken related to the frameworks to use. The selection was made according to their limitations, to the common objectives to achieve and adaptability of the frameworks to the ones that would be used by each student at the Desktop Application in order to accomplish their own personal objectives. Additionally, there was a need to identify which aspects of the previous version of the platform could be maintained and the ones that would need to be changed or adapted in order to be coherent with the new additions to the platform. All things considered, it is possible to include interactive scenarios in each game, customizing the actions which will trigger the transitions between them, the activities to be included in each scenario and associated rewards and penalties regarding player's score in game, as well as add resources to help complement the scenario information or be used as helps in game, among other customizable features. Also, in order to help users in getting familiar with the application, an Epik manual was created, which basically guides the user through the application, explaining in detail the available functionalities, how they work and how to use them.

Furthermore, the game generation process and how its information would be stored and processed by the Execution Environment had to be defined. After being validated, the game information is stored in a JSON file, where each of the Interactive Puzzle activities

have a specific section among the activities, in case the game contains any of this type of activity. This file suffered some changes along the dissertation in terms of its content and organization, as new features were being included and some improvements were being made during the process. When finishing developing, the game may be exported and later on imported to the Epik Website in order to be played by other users. Overall, most of the pretended objectives were accomplished for the Development Environment of the Epik platform. The most negative aspect about the Development Environment is the fact that the elements position in scenarios at the application sometimes is not saved, and when it happens the elements are displayed in random positions. After the development, the Desktop Application was submitted to an evaluation phase, where it was requested to a group of users to test the application in terms of the development of educational games and then answer to a survey to evaluate their experience. Overall, the reviews were positive, despite some issues with the simplicity in working with the platform, which tell that some improvements can be made. The survey section directed to the Interactive Puzzle activities also had positive reviews, in terms of ease of development, diversity of educational subjects to be integrated in the game context and overall inclusion in the Epik Games, which demonstrate that the puzzle activities have the potential to be used for learning purposes. However, some users found that the Development Environment of the crossword and jigsaw puzzles could have been more user friendly.

With regards to the development of the Execution Environment, the need to have a Website which would include information about the overall Epik platform, a download area for obtaining the Desktop Application and the activities plug-ins and a section dedicated to the Epik games was fulfilled. Regarding the last topic, developers are able to import their games and select which players are allowed to play the game by their email. Initially, the database stored most of the game properties and only part of it was provided by the game file. However, with the improvements made to the JSON file, as already referred in the beginning of this section, the game information was then obtained directly from the game file, meaning no database for most of the game properties and for each type of puzzle was required. This modification simplified the work to be done, since no queries were needed in order to access the database, and lowered the temporal complexity of the platform, making it more efficient. Thus, the database accesses are mostly in order to store and obtain logs from the game sessions played. Moreover, when in game, players may choose their name and avatar to be displayed, and their final score will be influenced by the game rules defined by the developer. After the game session ends, developers may see each player performance regarding that specific session, which allows them to have a feedback about each players' strengths and flaws. For this part of the common work there was no evaluation phase due to lack of time for preparing it correctly to the users. Unfortunately, some features related to the games execution were not fully developed, and will be discussed in the next section.

Taking into account that the development and inclusion of Interactive Puzzle activities to the Epik Platform and the overall common work developed is also a prototype, which

aims to evaluate the viability of game development and distribution, the next section 6.2 will present a set of improvements, modifications and new features that may be included as Future work, regarding Interactive Puzzle activities and also the Epik platform in general.

6.2 Future Work

This section presents some ideas for future work that may be developed within this dissertation's theme. There are some adjustments and improvements which can be done, with regards to the Interactive Puzzle activities and Epik platform implementation, usage and performance. Regarding the work developed in all the Epik platform, a future global improvement to seek is to make a deep analysis regarding temporal and spatial complexity of algorithms used in most functions. By reducing both complexities, the number of resources and memory used by the system would drop, making the platform more efficient, running smoother in computers with lower specifications. Addressing the Interactive Puzzle activities, follows a list of possible improvements and new features to be implemented in future versions:

- While playing a Multiplayer game, when in a puzzle activity, it would be interesting to have all players working on completing the same puzzle, instead of each one trying to complete it on their own;
- Instead of only allowing to have text or resource hints, specific helps for each type of puzzle could be included. Default hints could be implemented and the developer should choose if he wants to use them or not, regardless of other text or resource hints associated. For example, in the Photo Puzzle the correct position of a piece could be given or in the Crossword a letter in the right position could be provided as an hint as well;
- The scoring methods used for each type of puzzle activity only allows the developer to define the number of attempts or errors the players can make until their score % starts to drop from 100% and until they lost the game. Having this in mind, may be it would be interesting to implement the score algorithm in a different way that would give more freedom to the developer on the scoring method used;
- Regarding Crossword Puzzle, an interesting idea could be to implement a feature that would allow to have images or other resources as clues, instead of just text. Moreover, the Development Environment for it should be a bit simplified, once some users found it slightly confusing to interact with;
- About the Jigsaw Puzzle, the activity could allow the edition of more than just the text of the pieces and correspondent dropzones. An idea to consider would be to allow to modify the shape of the pieces and dropzones at the developers taste,

and even their display, instead of a grid. For example, create a Maths Pyramid by changing the display of the dropzones or create a jigsaw where each piece have a different shape and the corresponding dropzone has the same shape so they can fit on each other;

- Change the Puzzle builder system in order to allow an easier inclusion of new types of puzzle activities, since the current one is only prepared to have the three types developed.

About the Desktop Application, new features could be added and improved in order to make the development process easier and simpler. Some examples would be:

- When saving and exporting a project, identify locations to perform parallelism between Electron and SQLite3, in order to improve the performance of the platform due to the amount of queries associated to these functionalities;
- Include more shapes to the set of available ones, as well as a rotation feature for elements inside the scenarios and tools to allow undo and redo of changes made to the current project.
- Fix the issue of non adaptation of some components, due to their properties, to different screen resolutions. Also related to resolution, it could be interesting to have a feature for defining the screen resolution in game to allow an easier adaptation to players computers;
- A tool for project duplication, which would allow to reuse the structure of a project and all its content to create similar games. The same idea could be applied to copy scenarios within the same project and between projects, making it easier and faster to develop games;
- Feature to provide a way to organize games, projects, activities and resources by folders or categories, in order to improve elements organization in the workspace;
- Plug-ins system should be improved in order to allow to more easily include new type of activities in the future. Also regarding this topic, a revision of the game structure from the JSON files could be done, in case of inclusion of new types of activities to the games;
- JSON files structure can be improved, by not including parameters that are not used in all games. This aspect doesn't affect game performance, since only the actual game information is loaded, but may affect importation and exportation in a major scale, that is, when the game is more complex, since it increases the file size, which despite not being heavy enough to affect both processes, at some scale it can. Besides, improving the file structure will also help in terms of its organization and readability.

- Implement a limit of scenarios per game, content per scenario and restrict the resources properties to a certain level, in order to prevent the performance of the game execution from dropping and to also disallow incorrect practices at the development of the games.
- Modify some features of the application in order to adapt to the Linux version.

With regards to the Execution Environment, there is a need to execute performance tests to evaluate the server's behaviour when there are several sessions occurring at the same time, as well as when the game contains a bigger amount of scenarios, scenarios with a lot of resources or even resources that have properties which drop the platform performance. If problems are identified with game content, implementing limits on the game development, as told previously, or limiting the resources to be uploaded to the server to avoid lacking of space would prevent the performance of the game execution from dropping. That being said, a list of not fully developed features, improvements and new features to include in future work for the Execution Environment is displayed below:

- In Multiplayer games, the Scores scenario should also display the team score, which would be the sum of the scores of all the players in the session, as well as more information about the statistics of the players in game, such as the number of helps used, activities completed and respective scores and penalties, for both Single and Multiplayer.
- Helps were not included in the games execution. Context Resources helps and hints should be given by the system to the players, except in Multiplayer games, where hints would be provided by players which had already successfully completed a certain activity for which the help was requested. Additionally, each time a help was being given to a certain player, his state should be 'receiving help' and penalties for surpassing the limit of helps should have been given aswell. Also, players in the Finishers room could be able to help other players in completing certain activities when his help was requested;
- The Rankings scenario should be displayed at the end of Single Player games, showing the ranking regarding all the players from all the played sessions of the game. On Multiplayer games, this scenario should display the ranking regarding all teams involved in the played sessions of the game instead of the individual rankings of the players in the session. Another idea for both individual and team scores could be to create a section in the Website where both rankings would be displayed, regarding the top players and top teams for each game available publicly.
- The session logs should contain information about the players performance regarding all sessions of that game, for both Single and Multiplayer. Thus, more logs regarding players performance should be retrieved to the developer, such as the

scenarios and activities completed, the scores obtained in each of the completed activities, the penalties, if the player finish or lost the game, among others.

- Upon importing a game, it should be possible to add a previously edited list of emails regarding the players which are assigned to the game, instead of inserting the players emails one by one. Moreover, the game password should be sent to the emails of all the players assigned to play a game, as well as a notification email to warn them about the approaching of the end date for playing the game;
- Test if the platform works correctly in browsers other than the most used ones, since some features might not be working in the same way they do in other browsers;
- An Options menu during game execution might be implemented, in order to provide a way to pause the game (in Single player games), among other options that users may found useful;
- Include graphic visualization of session logs, so that the developer could observe more easily the statistics regarding specific players in certain game sessions;
- Organize public games in the platform by categories or teaching fields, in order to inform players about the game theme and to also allow searching for certain themes in their games.

An extra topic which can be interesting to take into consideration is the adaptation of these games, including Interactive Puzzle activities, to mobile devices, making them more available to the players since they can play them anywhere and any time, accelerating the learning process. The parameter creation suggestion in projects to allow screen resolution definition in game would also be useful for this adaptation to mobile devices.

Bibliography

[1] B. Sampaio, C. Morgado, and F. Barbosa. "Building Collaborative Quizzes." In: (2013).

[2] B. Sampaio. "Epik: Plataforma para Desenvolvimento de Jogos para Aprendizagem Colaborativa e Interativa." In: (2013).

[3] R. Pak. *Engaging Classroom Games for All Grades*. 2012. URL: http://www.teachhub.com/engaging-classroom-games-all-grades.

[4] Wikipedia. *Educational Game*. 2018. URL: https://en.wikipedia.org/wiki/Educational_game.

[5] R. Stathakis. *Five Reasons to Use Games in the Classroom*. 2013. URL: http://www.educationworld.com/a_curr/reasons-to-play-games-in-the-classroom.shtml.

[6] M. Pivec. "Editorial: Play and learn: potentials of game-based learning." In: *British Journal of Educational Technology* 38 (2007), pp. 387–393. DOI: 10.1111/j.1467-8535.2007.00722.x.

[7] J. Melero and D. Hernández-Leo. "A Model for the Design of Puzzle-based Games Including Virtual and Physical Objects." In: *Journal of Educational Technology and Society* 17.10 (2014), pp. 192–207. ISSN: 11763647, 14364522. DOI: 10.2307/jeductechsoci.17.3.192. URL: http://www.jstor.org/stable/jeductechsoci.17.3.192.

[8] J. R.-C. Kathryn Kane and Z. L. Berge. "Tapping into social networking: Collaborating enhances both knowledge management and e-learning." In: *VINE* 40 (2010), pp. 62–70. DOI: https://doi.org/10.1108/03055721011024928.

[9] P. Moreno-Ger, D. Burgos, I. Martínez-Ortiz, J. L. Sierra, and B. Fernández-Manjón. "Educational game design for online education. Computers in Human Behavior." In: *ELSEVIER* 24 (2008), 2530–2540. DOI: 10.1016/j.chb.2008.03.012.

[10] J. Davis. *Why Puzzles are so important for kids learning?* 2012. URL: https://www.learning4kids.net/2012/02/21/why-are-puzzles-so-good-for-kids-learning/.

[11] R. F. Mackay. *Playing to learn: Panelists at Stanford discussion say using games as an educational tool provides opportunities for deeper learning*. 2013. URL: https://news.stanford.edu/2013/03/01/games-education-tool-030113/.

[12] J. Wilson. *Puzzle*. US Patent 6,439,571. 2002.

[13] J. Hill, C. K. Ray, J. R. Blair, and C. A. Carver Jr. "Puzzles and games: addressing different learning styles in teaching operating systems concepts." In: *ACM SIGCSE Bulletin*. Vol. 35. 1. ACM. 2003, pp. 182–186.

[14] B. N. Verdine, R. M. Golinkoff, K. Hirsh-Pasek, and N. S. Newcombe. "Finding the missing piece: Blocks, puzzles, and shapes fuel school readiness." In: 3 (2014), pp. 7–13. DOI: https://doi.org/10.1016/j.tine.2014.02.005.

[15] C.-Y. Lin, H.-C. Chai, J. ying Wang, C.-J. Chen, Y.-H. Liu, C.-W. Chen, C.-W. Lin, and Y.-M. Huang. "Augmented reality in educational activities for children with disabilities." In: 42 (2016), pp. 51–54. DOI: https://doi.org/10.1016/j.displa.2015.02.004.

[16] M. van de Wouw. *Educational Games*. 2017. URL: https://www.digipuzzle.net/education/.

[17] Wikipedia. *Sudoku*. 2018. URL: https://en.wikipedia.org/wiki/Sudoku.

[18] Wikipedia. *Rubik's Cube*. 2018. URL: https://en.wikipedia.org/wiki/Rubik%27s_Cube.

[19] Google. *Google for Education > Blockly*. 2017. URL: https://developers.google.com/blockly/.

[20] Google. *Introcution to Blockly*. 2016. URL: https://developers.google.com/blockly/guides/overview.

[21] L. K. Group. *Scratch for Developers*. 2018. URL: https://scratch.mit.edu/developers.

[22] L. K. Group. *Snap! - Build your Own Blocks*. 2017. URL: http://snap.berkeley.edu/about.html.

[23] Scirra. *What is Construct2?* 2017. URL: https://www.scirra.com/construct2.

[24] Unity. *About us*. 2017. URL: https://unity3d.com/pt/public-relations.

[25] w3resource. *Node.js SQLite3*. 2018. URL: https://www.w3resource.com/node.js/nodejs-sqlite.php.

[26] GitHub. *Electron - Build cross platform desktop apps with JavaScript, HTML, and CSS*. 2019. URL: https://electronjs.org/.

[27] F. O. Source. *React - A JavaScript library for building user interfaces*. 2019. URL: https://reactjs.org/.

[28] W. W. W. Consortium. *HTML5 specification*. 2017. URL: https://www.w3.org/TR/html51/.

[29] W. W. W. Consortium. *CSS specification*. 2013. URL: https://www.w3.org/TR/css-style-attr/.

[30] Google. *Material-UI, React components that implement Google's Material Design*. 2019. URL: https://material-ui.com/.

[31] GitHub. *electron-builder, A complete solution to package and build a ready for distribution Electron, Proton Native or Muon app for macOS, Windows and Linux with "auto update" support out of the box*. 2019. URL: https://www.npmjs.com/package/electron-builder.

[32] W. W. W. Consortium. *The WebSocket API*. 2012. URL: https://www.w3.org/TR/websockets/.

[33] G. Rauch. *Socket.io*. 2019. URL: https://socket.io/.

[34] G. E. Krasner and S. T. Pope. "A cookbook for using the modelview controller user interface paradigm in smalltalk-80." In: *SIGPLAN Not.* 3 (1988), pp. 26–49.

[35] M. web docs. *File Reader*. 2019. URL: https://developer.mozilla.org/pt-PT/docs/Web/API/FileReader.

[36] GitHub. *axios - Promise based HTTP client for the browser and node.js*. 2019. URL: https://github.com/axios/axios.

[37] GitHub. *React Image Puzzle - Create a simple swap tile puzzle out of an image*. 2017. URL: https://www.npmjs.com/package/react-image-puzzle.

[38] A. Roworth. *How I Built a Pure CSS Crossword Puzzle*. 2017. URL: https://www.sitepoint.com/how-built-pure-css-crossword-puzzle/.

ANNEX I

USER TEST SURVEYS

I.1 Epik Games Development Application

- User Information

  1. What is your gender?
  2. How old are you?
  3. How many years have you been a teacher?
  4. What educational grade are you teaching?

- Games in Education

  5. What is your opinion about the use of games as a learning tool?
  6. Do you usually use games as a learning tool for your students?
  7. If yes, in which way?
  8. Have you ever used an application to create games as learning tool for your students?
  9. If yes, which applications have you used?

- Epik Interface

  10. On a scale from 1 to 5, how easy to use was the Application interface?
  11. How useful was the Epik Help Assistent in guiding you through the application, on a scale from 1 to 5 ?
  12. When generating a game, have you ever finished the process without success?

13. On a scale from 1 to 5, how clear was the error description?
14. Overall, on a scale from 1 to 5, how do you evaluate the interface organisation of the Epik Desktop Application?

- Epik Games - Scenarios and Flow

15. On a scale from 1 to 5, how do you rate the Epik games organization in scenarios?
16. Regarding the variety and utility of the available transitions for the scenarios, on a scale from 1 to 5, how do you rate them?
17. On a scale of 1 to 5, how do you rate the insertion/removal of resources in game scenarios?
18. On a scale from 1 to 5, how do you found the existing resources types (image, video, pdf and audio) in game scenarios useful ?

- Epik Games - Flow Component

19. Overall, on a scale from 1 to 5, how easy it is to create scenarios and associated flow?
20. On a scale from 1 to 5, how easy it is to create and delete a flow transition?
21. On a scale from 1 to 5, how easy it is to interact with the flow graph?

- Epik Games - Puzzle Activities

22. How easy was to develop the puzzle activities, on a scale from 1 to 5?
23. On a scale from 1 to 5, how useful do you think the inclusion of puzzle activities will be for games?
24. On a scale from 1 to 5, how do you evaluate the diversity of educational subjects that the 3 types of puzzle activities may provide?
25. Regarding the 3 types of puzzles available, how do you qualify each one of their development interfaces?

- Epik Games - Questions Activities

26. On a scale from 1 to 5, what is the probabilistic degree of using quizzes as educational activities in games?
27. Do you think that having 5 types of questions is enough to include a wide range of education subjects?
28. On a scale from 1 to 5, how easy was to develop questions activities?
29. Do you think that the tools and resources available to construct questions are enough to develop these type of activities?

- Epik Games - Collaboration and Helps

30. Overall, on a scale from 1 to 5, how do you rate the available collaboration mechanisms between players in the Epik Games?
31. Do you found the controlled chat, as a collaboration mechanism in an educational game, useful?
32. Do you found the free chat, as a collaboration mechanism in an educational game, useful?
33. How do you rate the notifications about player actions useful, on a scale from 1 to 5?
34. Do you found the display of player moods during an Epik game useful?
35. How do you rate, on a scale from 1 to 5, the utility of the hints in game?
36. How do you rate, on a scale from 1 to 5, the use of context resource in a game?
37. How do you rate, on a scale from 1 to 5, the collaboration between players in the Epik games?

- Satisfaction with the Epik Desktop Application

38. On a scale from 1 to 5, what is your degree of satisfaction with the Epik Desktop Application?
39. Would you use the Epik Desktop Application to build education games to be used on your classes?
40. If yes, in which way?
41. What words do you use to describe the Epik Application?
42. Would you recommend the Epik Application to your friends / colleagues?
43. If you have any suggestions, comments, or review please write them below.